



TM-(L)-HU-033/006/00

ANALYSES OF REQUIREMENTS FOR COMPUTER CONTROL
AND DATA PROCESSING EXPERIMENT SUBSYSTEMS

(NASA-CR-123572) ANALYSES OF REQUIREMENTS
FOR COMPUTER CONTROL AND DATA PROCESSING
EXPERIMENT SUBSYSTEMS. VOLUME 2: ATM
EXPERIMENT S-056 IMAGE (System Development
Corp.) 15 Mar. 1972 136 p

N72-23154

Unclas
15220
CSCL 09B G3/08

ATM EXPERIMENT S-056
IMAGE DATA PROCESSING SYSTEM
SOFTWARE DEVELOPMENT

VOLUME II

CR-123572



15 MARCH 1972

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

TECHNICAL MEMORANDUM

(TM Series)

This document was produced by SDC in performance of contract NAS8-25471

ATM EXPERIMENT

S-056

IMAGE DATA PROCESSING SYSTEM

SOFTWARE DEVELOPMENT

VOLUME II

15 MARCH 1972

SYSTEM

DEVELOPMENT

CORPORATION

4810 BRADFORD BLVD., N. W.

HUNTSVILLE

ALABAMA

35805



TABLE OF CONTENTS

	<u>Page No.</u>
FORWARD	v
1. <u>INTRODUCTION</u>	1-1
1.1 INSTRUCTIONS FOR USING IDAPS	1-1
1.2 DECK SET-UP FOR LOADING AND OPERATION	1-3
1.2.1 <u>Operation from Card Deck</u>	1-4
1.2.2 <u>Operation from IBSYS FAST TAPE (B5)</u>	1-4
2. <u>OPERATORS</u>	2-1
2.1 AVERAGE.	2-4
2.2 INTEGRATE.	2-5
2.3 SCALE.	2-6
2.4 INVARIANT ALTER.	2-7
2.5 DEPENDENT ALTER.	2-8
2.6 CONVOLVE	2-9
2.7 DECONVOLVE	2-10
2.8 SIZE	2-12
2.9 COPY	2-13
2.10 TRANSPOSE.	2-14
2.11 EXPAND	2-15
2.12 CHANGE FORMAT.	2-16
2.13 DISPLAY FORMAT	2-17
2.14 STOP	2-18
2.15 RESOLUTION CHART	2-19
2.16 PRINT.	2-20
2.17 PSF GENERATOR.	2-22
2.18 EXTRACT.	2-23
2.19 FORCON (Fourier Convolution)	2-24
2.20 CORRELATE.	2-25
2.21 FFT (Fast Fourier Transformation).	2-26
2.22 IFFT (Inverse Fast Fourier Transformation)	2-27

TABLE OF CONTENTS (continued)

	<u>Page No.</u>
2.23 FEATHER	2-29
2.24 FIXED TO FLOAT	2-30
2.25 FLOAT TO FIXED	2-31
2.26 ASSIGN	2-32
2.27 RELEASE	2-34
2.28 COMMENT	2-35
2.29 PAUSE	2-36
2.30 VARIABLE	2-37
2.31 MACRO	2-39
2.32 STACK	2-40
2.33 MULTIPLE DISPLAY	2-41
2.34 LABEL	2-42
2.35 CENTER TRANSFORM	2-44
2.36 SPLIT	2-45
2.37 MERGE	2-46
2.38 M/P TO R/I	2-47
2.39 R/I TO M/P	2-48
2.40 LOG MAGNITUDE	2-49
2.41 GRADIENT	2-50
2.42 REWIND	2-51
2.43 MATH	2-52
2.44 OPERATOR SUMMARY	2-54
 3. <u>SOFTWARE</u>	 3-1
IDAPS	3-2
DATAIN	3-3
PACK	3-4
UNPACK	3-5
UNPAC	3-6
MAC	3-7
ASSGN	3-8
FREE	3-9

TABLE OF CONTENTS (continued)

	<u>Page No.</u>
VARY	3-10
CONVRT	3-11
MLTAVS	3-13
SPINT.	3-14
EPAND.	3-16
CNVOL.	3-17
SHADE.	3-18
PDGSA.	3-19
DISPLAY.	3-21
TRNPS.	3-22
RSLGEN	3-23
PSFGEN	3-24
HRDCPY	3-25
CONV	3-26
DCONV.	3-29
CRLATE	3-30
XFORM.	3-31
IXFORM	3-32
FRX2V.	3-33
FRXFM.	3-40
SPRED1	3-45
SPRED2	3-46
SPRED3	3-48
SHAVE.	3-49
FLTOFX	3-50
FXTOFL	3-51
STKPIC	3-52
MULDSP	3-53
LABL	3-54
DSPLIN	3-56
CENTR.	3-57

TABLE OF CONTENTS (continued)

	<u>Page No.</u>
SPLT	3-58
MERG	3-59
MPTORI	3-60
RITOMP	3-61
LOGMAG	3-62
PDAFNL	3-63
TCOPY.	3-65
ARITH.	3-66
SCAL	3-67

LIST OF FIGURES

Figure 1	IBM 7094 Core Allocation	1-2
Figure 2	Deck Set-Up for FAST TAPE Operation Without Subroutine Editing	1-5
Figure 3	Deck Set-Up for FAST TAPE Operation With Subroutine Editing.	1-6
Figure 4	Flow of Subroutine CONV.	3-28
Figure 5	128 x 128 Data Storage Layout.	3-38
Figure 6	Subroutine FRX2V Simplified Flow	3-39

15 March 1972

-v-

TM-(L)-HU-033/006/00

FORWARD

This document is the second of two produced under NASA contract Number NAS8-25471, "Analyses of Requirements for Computer Control and Data Processing Experiment Subsystems." The report was prepared by the System Development Corporation's (SDC) Huntsville Space Projects staff for the Computer Systems Division of the George C. Marshall Space Flight Center's Computation Laboratory. Volume I, ATM Experiment S-056, Image Data Processing System/Technique Development, presents the results of the image processing studies for S-056 pictures and Volume II, ATM Experiment S-056, Image Data Processing System/Software Development, documents the software designed and implemented by SDC in support of the image processing studies.

This work was performed under the technical direction of Mr. Bobby C. Hodges, Contracting Officer's Representative for the project. Appreciation is expressed to Mr. Hodges, Mr. Doug Thomas and Dr. E. H. Hopper of the MSFC Computation Laboratory, and to Mr. J. E. Milligan of Space Sciences Laboratory for their support and technical assistance during the course of this project. We would also like to thank Mr. A. G. Gann of CSC for supplying the assembly language routine, PUPINS.

SECTION 1. INTRODUCTION

IDAPS (Image Data Processing System) is a user oriented, computer based, language and control system for use in processing image data. The purpose of the system is:

- To provide a framework or standard for implementing image data processing applications.
- To simplify set-up of image processing runs to a point that the system may be used without a working knowledge of computer programming or operation.
- To streamline operation of the image processing facility.
- To allow multiple applications to be run in sequence without operator interaction.

The IDAPS control system loads the IDAPS operators, interprets the input, constructs the necessary parameters for each application, and calls the application. The overlay feature of the IBSYS loader (IBLDR) provides the means for running multiple operators which would otherwise overflow core storage. Figure 1 presents the core storage allocation scheme.

IDAPS has been intentionally designed to be simple in concept and easy to modify. As new requirements are established, the system can be modified to include additional features and capabilities.

1.1 INSTRUCTIONS FOR USING IDAPS

The IDAPS system language consists of basic application operators such as AVERAGE, CONVOLVE, PRINT, CHANGE FORMAT, etc., and system operators such as SIZE, ASSIGN, PAUSE, STOP, etc. to control the execution of the application operators.

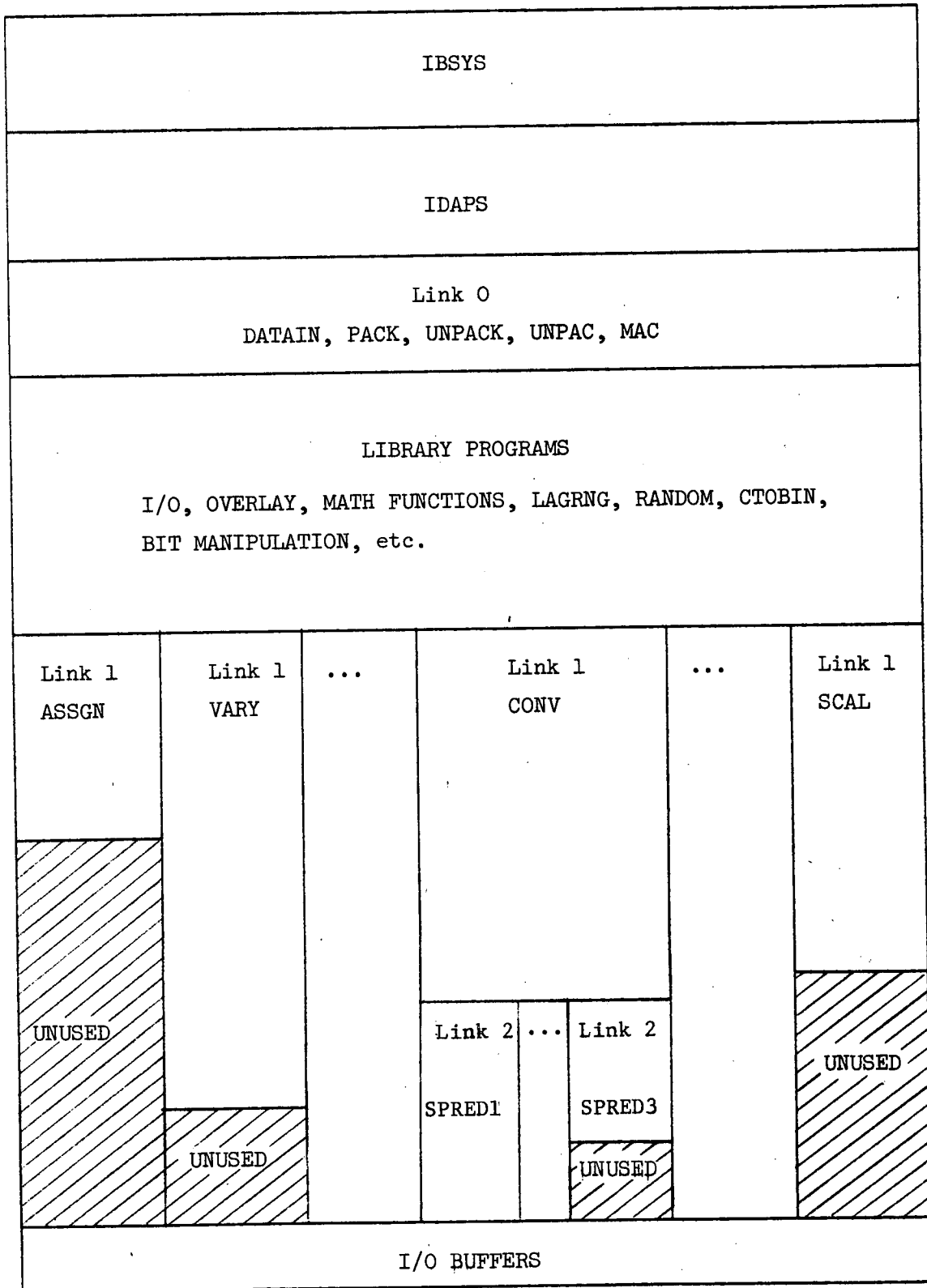


Fig. 1 - IBM 7094 Core Allocation

There are two basic modes for operating under IDAPS:

- STEP-BY-STEP MODE
- ZIP MODE

The STEP-BY-STEP MODE is the most direct method for using IDAPS. In this mode the operator cards are simply stacked in the card reader and all tape assignments and procedural requirements are made by the system. Operating instructions are printed on the on-line printer. Since IDAPS assigns input, output and scratch tapes in accordance with a prescribed order, this mode may require considerable tape manipulation and operator interaction and consequently is subject to operator mistakes. This mode is most useful where short runs are desired or where runs consist of only a few operators. The beginning user of IDAPS may prefer this mode since automatic program stops occur after each instruction, thus providing a better chance to keep up with what's going on.

The ZIP MODE of operation is initiated by depressing sense switch 4 and pressing START. Before using the ZIP MODE it is important to set up the IDAPS operator deck with ASSIGN and RELEASE cards so that proper input and output tape assignments are made by the system. A careful review of the normal tape assignment list (Paragraph 2.), the ASSIGN operator (Paragraph 2.26), and the RELEASE operator (Paragraph 2.27) should be made before attempting to operate in the ZIP MODE. To return from ZIP into the STEP-BY-STEP MODE, simply put sense switch 4 in the normal up position. The primary advantage of the ZIP MODE is that it gives the user the ability to specify very long and complex sequences which can be run with a minimum of operator interaction.

1.2 DECK SET-UP FOR LOADING AND OPERATION

IDAPS is designed to take advantage of the 7094 IBSYS operating system and standard library routines. Because the system has been designed for simplicity and speed in using operational routines while at the same time providing a means for testing and installing new routines, two different methods of operation may be used, i.e.:

- Operation from Card Deck
- Operation from IBSYS FAST TAPE (B5)

1.2.1 Operation from Card Deck

Original FORTRAN and binary decks are maintained by the System Development Corporation for use in periodic major updates of the system. Because of the size of these decks and the time required for on-line card reading, the use of these decks is not recommended as a general method for program loading by the IDAPS user.

1.2.2 Operation from IBSYS FAST TAPE (B5)

An up-to-date compilation of IDAPS is maintained by the System Development Corporation on tape drive B5 as a part of the facilities program library (FAST TAPE). By loading a small control deck into the computer, it is thus possible to bring the IDAPS system on-line and to modify or add portions of the system. Certain IDAPS operators are intended for use where extensive modifications are expected, i.e., MACRO and VARIABLE. These operators may also be revised by including the proper IBSYS EDIT cards in the control deck.

Figure 2 illustrates the deck set-up for calling IDAPS from the FAST TAPE library if there is no editing to be done to any of the operators within IDAPS. This deck is available at the computer facility along with the library tape for those who wish to use IDAPS in this manner.

Figure 3 illustrates how the FAST TAPE call deck is modified if changes are desired in some parts of IDAPS. In this case subroutine MAC and subroutine VARY are modified.

Once the FAST TAPE card deck is set up, operation under IDAPS proceeds as follows:

- 1) Mount, load, and ready scratch tapes on drives A3, A4, B2, B3, and B4, and a system print tape on B1.
- 2) Mount, load, and ready the IBSYS system tape on A1, and the IBSYS FAST TAPE on B5.

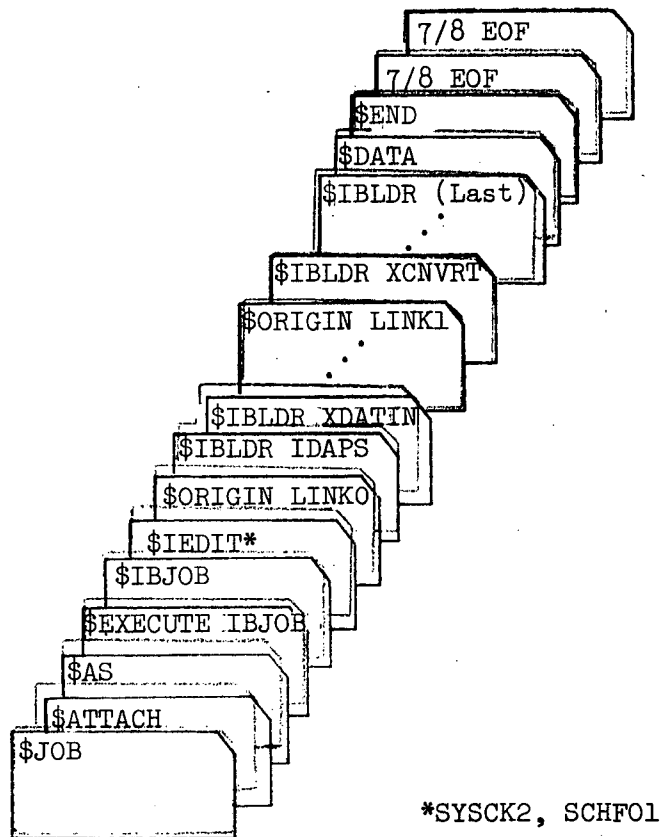


Figure 2 - Deck Set-up for FAST TAPE Operation
Without Subroutine Editing

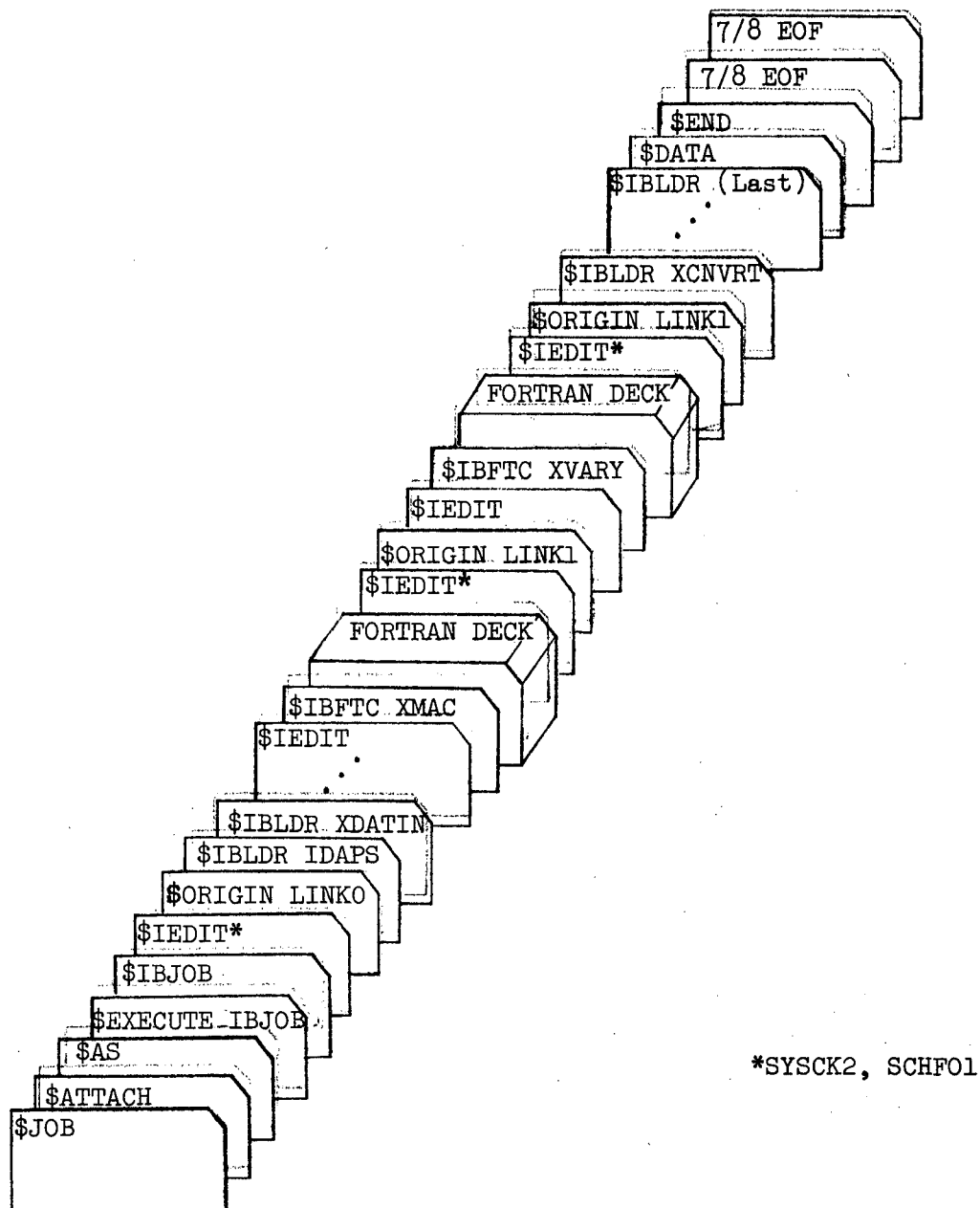


Figure 3 - Deck Set-up for FAST TAPE Operation
With Subroutine Editing

15 March 1972

1-7

TM-(L)-HU-033/006/00

- 3) Clear and reset the computer.
- 4) Clear the card reader.
- 5) Place sense switch 1 down, all others up.
- 6) Load the FAST TAPE card deck and press the card reader START to initialize.
- 7) Press LOAD TAPE.
- 8) After IDAPS is loaded and compiled, place sense switch 1 up and then follow instructions of the on-line printer.
- 9) In the STEP-BY-STEP MODE, complete operating instructions are printed on-line. If ZIP MODE is initiated by putting sense switch 4 down, no regular program stops occur.

SECTION 2. OPERATORS

A fixed field format is used for specifying IDAPS operators. The general form is:

Col 1	Col 24	Col 30	Col 36	
OPERATOR NAME	Parameter #1	Parameter #2	Parameter #3	Param #4
	6 Columns 6 Columns 6 Columns			

The operator name starts in column 1. Since only the first three characters of each operator will be examined, each operator may be abbreviated to three or more characters. Parameters, if required, are generally six digit integers and are right justified to columns 24, 30, 36, 42, 48, 54, 60, 66, and 72.

Where data cards are used for entering lists of data, a free form format is used. The data name begins in column 1, followed by an equal sign and the string of desired data, i.e.,

CURVE = 63.,31.5,16.,12.3,7.2,4.1,3.2,2.1,1.3,.7\$

The list is terminated with a dollar sign. If data extends beyond column 78, a continuation card(s) is required. Column 1 of a continuation card must be a blank, i.e., data begins in column 2.

With the exception of those IDAPS operators which accept data from or prepare data for the image processing scan/display equipment, all data is in standard FORTRAN tape format with each picture point occupying a computer word. Data is stored and manipulated in both fixed point and floating point with floating point being used where accuracy is needed and fixed point (integers) for its simplicity and compatibility with scanner/display equipment.

IDAPS assigns tape units, as needed, from a fixed list of available drives. As each unit is assigned a task, as a part of an operator (input, output, or scratch), it is flagged and the next assignment is made to the next un-flagged tape unit on the list. At the end of the execution of an IDAPS operator all tape units assigned by IDAPS are released and the assignment flags are removed. By using the ASSIGN operator (Paragraph 2.26) and the RELEASE operator (Paragraph 2.27) it is possible to place assignment flags by any tape unit on the assignment list as desired. Thus, if the first unit on the list is ASSIGN'ed, IDAPS will skip over that drive when making assignments for an operator. The RELEASE operator removes desired assignment flags and makes the RELEASE'd unit available for assignment by IDAPS.

To assist the user in using ASSIGN and RELEASE, all operators assign tapes in the following order:

INPUT
OUTPUT
SCRATCH

The following is the list of tape units in the order assigned by IDAPS:

<u>Logical Unit Number</u>	<u>Tape Drive Number</u>
8	A5
10	A6
12	A7
14	A9
11	B6
13	B7
15	B9

Following IDAPS compilation, tape drives A3, A4, and B4 are no longer needed by IBSYS and these drives may be used for normal tape unit assignment by IDAPS.

An effort has been made to follow standard FORTRAN notation with regard to fixed or floating point notation for parameter names. Thus, integers will be noted by names beginning with the letters I, J, K, L, M, or N, and floating point numbers by names beginning with other letters.

Reference to a frame of data means a two dimensional array of numbers which usually represents the varying gray levels of an image. A single number within a frame of data is referred to as an element, picture element, or PIXEL, and is addressed by its line and column number.

Because of the tremendous amount of data manipulation required by many image processing operations, care should be taken to restrict the size of frames to be processed. Generally, processing time goes up as the square of the size of the frame being processed and very long runs are generated when processing frames greater than 256 by 256 elements.

IDAPS reads operators in BCD form and converts the parameters to integer or floating point by using MSFC library routine CTOBIN (BCD to Binary Conversion). Due to the real-time conversion of FIOS during execution, the card reader is selected on the first call to CTOBIN. This requires a non-executable IDAPS operator card following the first IDAPS card containing integer or floating point parameters. The suggested IDAPS operator card setup is as follows:

```
SIZE           N      M
COMMENT
                (Next Operator)
```

To specify a two dimensional function in an IDAPS operator (SIZE, EXPAND, PRINT, etc.) the first variable specified corresponds to the line or vertical dimension and the second refers to the column or horizontal dimension. For example, in the paragraph above, the SIZE operator requires the specification of the integer variables N & M. In this case the (N) specifies the number of lines in the picture array, and the (M) specifies the number of columns.

2.1 AVERAGE Operator

The AVERAGE operator specifies that "n" frames are to be averaged--picture element by picture element.

Col. 1	Col. 24
AVERAGE	N

N indicates the number of frames to be averaged.

Normal tape assignments:

A5	} Input (N tapes will be assigned for input)
A6	
°	
°	
°	

Output (the Nth +1 tape will be assigned for output)

Data type:

Input - Integer

Output - Integer

Subroutines:

MLTAVS

2.2 INTEGRATE Operator

The INTEGRATE operator spatially integrates each element of a frame by averaging its eight neighboring elements and adjusting the center element by some percentage to bring it closer to the average of its neighbors.

Col. 1	Col. 24	Col. 30
INTEGRATE	L	Y

L is an integer value which specifies the limit over which all values will be adjusted.

Y is a floating point value which specifies the percentage by which the element will be adjusted.

L and Y are right justified to columns 24 and 30 respectively.

Normal tape assignments:

A5 - Input

A6 - Output

Data Type:

Input - Floating point

Output - Floating point

Subroutines:

SPINT

2.3 SCALE Operator

The SCALE operator scales floating point data between 0 and 63.

Col. 1	Col. 24	Col. 30	Col. 36	Col. 42	
SCALE	K	L	A	B	

K specifies the type of scaling.

K = 1 - Logarithmic

K = 2 - Square root

K = 3 - Cube root

K = 4 - Square

K = Other - linear

L is clipping flag

L = 1 - Clip outside of A and B

L = Other - No clipping

A is the lower clipping level -- all input values below A will be set to 0.

B is the upper clipping level -- all input values above B will be set to 63.

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Floating point

Output - Floating point

Subroutines:

SCAL

2.4 INVARIANT ALTER Operator

The INVARIANT ALTER operator alters the gray scale value of the picture elements based on the original gray scale value, independent of position within the frame.

Col.
1

INVARIANT ALTER

The table of alteration values (table K) is input immediately following the INVARIANT ALTER operator as follows:

Col.
1*

K=63,62,61,...1,0\$

Table K is 64 words long and is ordered from gray scale of 0 to gray scale of 63, i.e., the value in the first position of table K will replace all original elements whose gray scale value is 0. In the above example, elements with gray scale of 0 would be changed to 63, 1 to 62, etc.

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer

Output - Integer

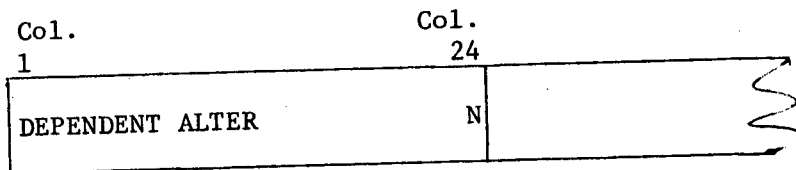
Subroutines:

SHADE

*Continuation cards begin in column 2.

2.5 DEPENDENT ALTER Operator

The DEPENDENT ALTER operator performs a position dependent alteration of the gray scale value of each picture element.



N specifies the length of Tables A and B. The values of the A and B tables represent the coefficients of the linear expression $A + BX$, where X is the original picture element value. The index of the A and B tables is determined by the radial distance from the center of the frame to the element. These tables are input immediately following the DEPENDENT ALTER operator. For example, if N is 10, the A and B tables may be input as follows:

```
Col.
1*
A=3*0.,2*1.,2.,3.,5.,8.,12.$
B=3*1.,.9,.7,.6,.5,.45,.42,.42$
```

Normal tape assignments:

```
A5 - Input
A6 - Output
```

Data type:

```
Input - Integer
Output - Integer
```

Subroutines:

```
PDGSA
```

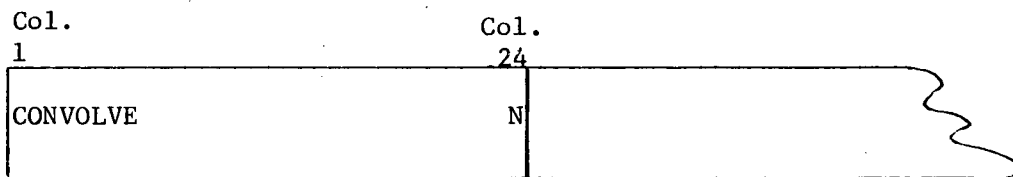
Restrictions:

N must be equal to or greater than one half the diagonal length of the picture to be processed.

*Continuation cards begin in column 2.

2.6 CONVOLVE Operator

The CONVOLVE operator numerically convolves a frame of data with a two-dimensional PSF.



N is the size of the PSF which is input from tape.

Normal tape assignments:

A5 - Input image

A6 - Input PSF

A7 - Output

Data type:

Input - Integer

Output - Integer

Subroutines:

CNVOL

Restrictions:

(1) Image size is limited to 128 x 128.

(2) PSF size is limited to 63 x 63.

2.7 DECONVOLVE Operator

The DECONVOLVE operator employs the Fast Fourier Transform (FFT) to deconvolve a frame of data with a point spread function (PSF). It divides frequency terms of the PSF into corresponding image terms and then takes the inverse transform.

Col. 1	Col. 24	Col. 30	Col. 36	Col. 42	Col. 48	Col. 54	Col. 60	Col. 66
DECONVOLVE	N	M	K	J	L	I	A	B

N is the size of the PSF.

M is the size of the deconvolved image.

K is a flag to indicate whether the PSF is input in the spatial or frequency domain.

K \neq 1 - Spatial

K = 1 - Frequency

J is the limit on magnitude amplification.

J = blank - no limit*

J = 0 - allow no magnitude amplification

J = other - limit amplification to a maximum of J.

L is the limit on phase magnification specified in degrees.

L = blank - no phase limit

L = 0 - allow no phase amplification

L - other - limit phase change to a maximum of L.

I is a flag to indicate clipping.

I = blank - clip all values outside of 0. and 63.

I = 1 - no clipping

I = 2 - clip all values outside A (maximum) and B (minimum)

A is the maximum clipping value

B is the minimum clipping value

Normal tape assignments:

A5 - Input image

A6 - PSF

A7 - Output image

A9 }
B6 } Scratch
B7 }

B9 - Output - PSF in frequency domain if $K \neq 1$. If $K = 1$, B9 is not used.

Data type:

Input - Floating point

Output - Floating point

Subroutines:

DCONV

FRX2V

FRXFM

SPRED2

Restrictions:

- 1) The size of the deconvolved image must be a power of 2 equal to or greater than the size of the input image.
- 2) The size of the PSF must be equal to or less than the size of the input image.

*When the magnitude of a point of the PSF transform is zero the corresponding point of the image transform is unaltered (J = blank).

2.8 SIZE Operator

The SIZE operator is intended to specify the size of the input frame to be processed by the successor operators. A new size specification need not be made until there is a change in the size of the frame.

Col. 1	Col. 24	Col. 30
SIZE	M	N

M represents the number of lines per frame.

N represents the number of columns per frame.

2.9 COPY Operator

The COPY operator copies 1 file from one tape to another.

Col. 1	Col. 24	Col. 30	Col. 36	Col. 42	Col. 48	
COPY	I	J	K	L	M	

I is the input tape logical unit number.

J is the output tape logical unit number.

K is the file number of the file on the input tape to be transferred.

L is the file number on the output tape.

M is the Interleaved/Non-Interleaved FLAG.

M = 1 Interleaved real/imaginary or magnitude/phase data is to be copied.

= 2 Scanner format data is to be copied.

= other Standard copy.

Normal tape assignments:

(None - must be assigned by user)

Data type:

Input - Floating point or integer

Output - Floating point or integer

Subroutines:

TCOPY

2.10 TRANSPOSE Operator

The TRANSPOSE operator transposes a frame of image data so that the n th column becomes the n th line.

Col.

1

TRANSPOSE



Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer

Output - Integer

Subroutines:

TRNPS

Restrictions:

- (1) The size of the frame of data to be transposed must be less than or equal to 256 x 256.

2.11 EXPAND Operator

The EXPAND operator can be used to identify a square area of a frame of data and to expand that area by some whole number factor.

Col. 1	Col. 24	Col. 30	Col. 36	Col. 42
EXPAND	I	J	N	K

I, J specifies the upper left hand line and column number of the square area to be expanded.

N is the size of the area to be expanded.

K indicates the expansion factor.

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer

Output - Integer

Subroutines:

XPAND

2.12 CHANGE FORMAT Operator

The CHANGE FORMAT operator converts from scanner tape format to FORTRAN tape format.

Col. 1	Col. 24	Col. 30	Col. 36	
CHANGE FORMAT	M	N	J	

M is the starting line number.

N is the starting column number.

J is the file number of the frame.

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer (scanner format)

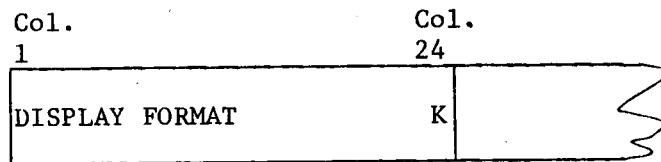
Output - Integer (FORTRAN format)

Subroutines:

CONVRT

2.13 DISPLAY FORMAT Operator

The DISPLAY FORMAT operator converts from FORTRAN tape format to the format of the display device.



K is a flag to superimpose a 32 x 32 grid on frame of data.

K = 1 - Grid

K = other - No grid

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer

Output - Integer (Display format)

Subroutines:

DSPLAY

2.14 STOP Operator

The STOP operator terminates the run sequence and returns control to IBSYS.

Col.

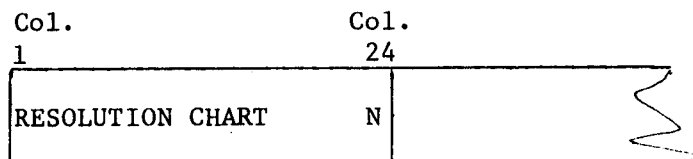
1



STOP

2.15 RESOLUTION CHART Operator

The RESOLUTION CHART operator generates a resolution chart of vertical bars.



N specifies the size of the square resolution chart.

Normal tape assignments:

A5 - Output

Data type:

Output - Integer

Subroutines:

RSLGEN

Restrictions:

- (1) The size of the resolution chart must be a power of 2 less than or equal to 2048.

2.16 PRINT Operator

The PRINT operator prints a portion of a frame of data.

Col. 1	Col. 24	Col. 30	Col. 36	Col. 42	Col. 48	
PRINT	J	N	M	L	K	

J is the print option

- J = 64 - Gray level overprint
- J = -1 - Gray level values (integer - off-line)
- J = -2 - Gray level values (integer - on-line)
- J = -3 - Floating point (off-line)
- J = -4 - Floating point (on-line)
- J = 0-63 - Bilevel - blank will be printed for values $\leq J$, X for those $> J$.

N, M is the upper left corner of area to be printed.

L, K is the size of the area to be printed.

Integer values are printed with a maximum of 32 numbers per print line. If the variable (K) is greater than 32, the PRINT operator will print the first 32 columns of all lines specified and then print the next 32 columns in a second pass. As many printing passes as necessary will be made to print the area specified by L and K.

Gray level overprint (J=64) and bilevel print options are handled in the same manner except that a maximum of 132 columns are printed with each pass.

Normal tape assignments:

A5 - Input

Data type:

Input - Integer or floating point depending on input specifications.

15 March 1972

2-21

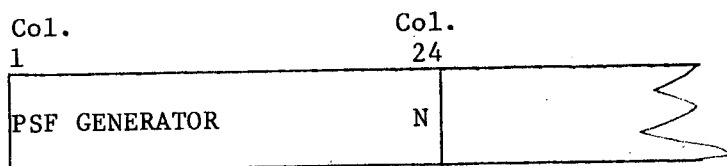
TM-(L)-HU-033/006/00

Subroutines:

HRDCPY

2.17 PSF GENERATOR Operator

The PSF GENERATOR generates an $N \times N$ point spread function and stores it on tape.



N is the size of the PSF.

The PSF data is input as follows:

Col

1*

CURVE = 63.,30.,10.,5.\$

Normal tape assignments:

A5 - Output

Data type:

Output - Floating point

Subroutines:


PSFGEN

Restrictions:

The size of the PSF generated (N) should be no greater than 100.

2.18 EXTRACT Operator

The EXTRACT operator extracts a portion of a frame of data.

Col. 1	Col. 24	Col. 30	Col. 36	
EXTRACT	I	J	N	

I, J specifies the upper left hand line and column number of the square area to be extracted.

N is the size of the square area to be extracted.

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer

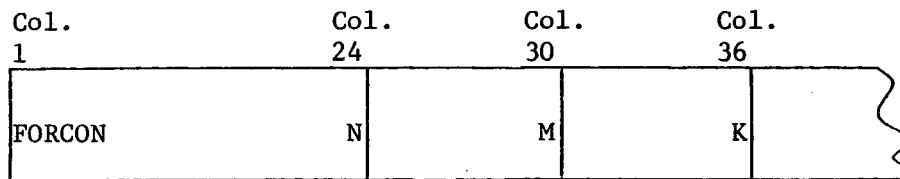
Output - Integer

Subroutines:

XPAND

2.19 FORCON Operator

The FORCON operator convolves an image using the Fast Fourier Transform.



N is the size of the point spread function.

M is the size of the convolved image.

K is a flag to indicate whether the PSF is to be input in the spatial or frequency domain.

K \neq 1 - Spatial

K = 1 - Frequency

Normal tape assignments:

A5 - Input image

A6 - PSF

A7 - Output image

A9 }
 B6 } Scratch
 B7 }

B9 - Output - PSF in frequency domain if K \neq 1. If K = 1, B9 is not used.

Data type:

Input - Floating point

Output - Floating point

Subroutines:

CONV

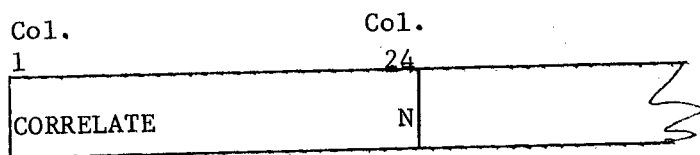
FRX2V

FRXFM

SPRED1

2.20 CORRELATE Operator

The CORRELATE operator correlates two images.



N is the size of the correlated output array.

Normal tape assignments:

A5 - Input image 1

A6 - Input image 2

A7 - Output image

A9	} Scratch
B6	
B7	
B9	

Data type:

Input - Floating point

Output - Floating point

Subroutines:

CRLATE

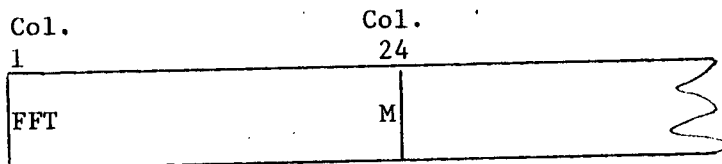
FRX2V

FRXFM

SPRED3

2.21 FFT Operator

The FFT (Fast Fourier Transform) operator transforms an $N \times N$ image to produce $M \times M$ real and imaginary arrays in the frequency domain.



M is the size of the transformed array.

Normal tape assignments:

A5 - Input
A6 - Output
A7 } Scratch
A9 }

Data type:

Input - Floating point
Output - Floating point (interleaved line by line)

Subroutines:

XFORM
FRX2V
FRXFM

Note: Because the FFT output tape is an interleaved combination of the real and imaginary components of the transformed image, its size is $2M \times M$ elements and subsequent operations on the output tape must make allowance for this size change.

2.22 IFFT Operator

The IFFT (Inverse Fast Fourier Transform) operator converts an array of real and imaginary components in the frequency domain to an image array.

Col. 1	Col. 24	Col. 30	Col. 36	Col. 42	
IFFT	M	I	A	B	

M is the size of the output image array.

I is a flag to indicate clipping.

I = blank - clip outside 0. and 63.

I = 1 - no clipping

I = 2 - clip outside A and B

A is the maximum clipping value.

B is the minimum clipping value.

Normal tape assignments:

A5 - Input

A6 - Output

A7 } Scratch
A9 }

Data type:

Input - Floating point (interleaved line by line)

Output - Floating point

Subroutines:

IXFORM

FRX2V

FRXFM

15 March 1972

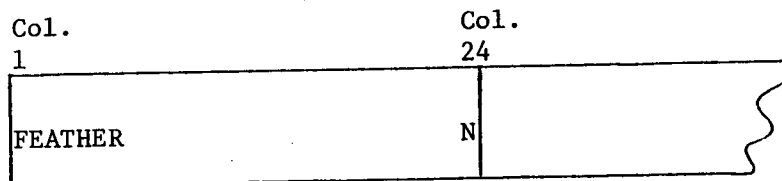
2-28

TM-(L)-HU-033/006/00

Note: Because the IFFT operator requires an input tape which is the interleaved combination of real and imaginary components, its size is 2MXM and must be appropriately specified, otherwise only half the needed number of records will be read.

2.23 FEATHER Operator

The FEATHER operator rounds off the edges of a frame of data.



N is the size of the modification table which is input as follows:

Col.
1*

MOD=.1,.3,.5,...\$

Where the first value corresponds to the extreme edge of the frame of data to be feathered.

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer

Output - Integer

Subroutines:

SHAVE

Restrictions:

No more than 25 values should be listed for MOD.


*Continuation cards begin in column 2.

2.24 FIXED TO FLOAT Operator

The FIXED TO FLOAT operator converts a tape in fixed point to floating point.

Col.

1



FIXED TO FLOAT

Normal tape assignments

A5 - Input

A6 - Output

Data type:

Input - Integer

Output - Floating point

Subroutines:

FXTOFL

2.25 FLOAT TO FIXED Operator

The FLOAT TO FIXED operator converts a tape in floating point to fixed point.

Col.

1

FLOAT TO FIXED

Normal tape assignments

A5 - Input

A6 - Output

Data type:

Input - Floating point

Output - Integer

Subroutines:

FLTOFX

2.26 ASSIGN Operator

The ASSIGN operator is used to artificially assign a tape unit. The purpose of this operator is to avoid manual tape unit changes during operation by allowing the system user to block out specified tape units so that tape assignments for succeeding operators will not include those that have been artificially assigned.

Col. 1	Col. 24	Col. 30	Col. 36	
ASSIGN	N	M1	M2	etc.

N is the number of tape units to be artificially assigned.

M1, M2, ... are the logical tape unit numbers.

Example: Suppose the following set of operators are executed:

DECONVOLVE	...
EXTRACT	...
DISPLAY FORMAT	...

Under normal IDAPS operation tape units are assigned in the following order: A5, A6, A7, A9, B6, B7 and B9 (logical units 8, 10, 12, 14, 11, 13 and 15). Thus, for the operator DECONVOLVE, the input image is assigned unit A5, the PSF is assigned A6 and the output is generated on tape unit A7.

The input tape assigned by IDAPS for EXTRACT is A5 and the output tape is A6. Likewise, the input tape for DISPLAY FORMAT is A5 and the output tape is A6. This involves a manual change of tape units between each operator. To avoid this manual change of tape units, the following set of operators may be used:

DECONVOLVE	...			
ASSIGN	2	8	10	
EXTRACT	...			
ASSIGN	1	12		
DISPLAY FORMAT	...			
RELEASE*	3	8	10	12

The output of the DECONVOLVE operator is on A7 (logical unit 12). The first ASSIGN operator prohibits tapes A5 (logical unit 8) and A6 (logical unit 10) from being used by succeeding operators; thus, operator EXTRACT is assigned the next available tape unit for its input, i.e., A7 (logical unit 12) which is the unit on which the output of the DECONVOLVE operator was written. The EXTRACT operator output is generated on A9 (logical unit 14). Since unit A9 now contains the desired input data for the DISPLAY FORMAT operator, it is necessary to artificially assign unit A7 (logical unit 12), thus causing IDAPS to assign unit A9 as the input unit for the DISPLAY FORMAT operator. Finally, those tape units which have been artificially assigned are released for system use by using the RELEASE* operator. Using the above set of operators, no manual tape unit changes are necessary.

2.27 RELEASE Operator

The RELEASE operator is used to release tape units artificially assigned by operator ASSIGN*.

Col. 1	Col. 24	Col. 30	Col. 36	
RELEASE	N	M1	M2	etc.

N is the number of tapes to be released.

M1, M2, ... are the logical unit numbers.

*See Section 2.26.

2.28 COMMENT Operator

The COMMENT operator provides instruction or identification messages on both the on-line and off-line printers.

Col.

1

COMMENT (Any Message)

Columns 1 through 78 (including the operator name, COMMENT) are printed.

2.29 PAUSE Operator

The PAUSE operator initiates a halt in system operation and prints a message on the on-line printer.

Col.

1

PAUSE (Any Message)

Columns 1 through 78 are printed.

Operation is renewed by pressing START on the operator console.

2.30 VARIABLE Operator

The VARIABLE operator is used to insert temporary, one-shot type subroutines in the system without making system modifications.

Col.

1

VARIABLE (Any Desired Parameters)

This operator will initiate a call to subroutine VARY. The SUBROUTINE statement in subroutine VARY will be as follows:

SUBROUTINE VARY (NL, NC, CARD)

NL - Number of lines from preceding SIZE operator

NC - Number of columns from preceding SIZE operator

CARD - An array containing the image of the VARIABLE operator card which may contain any desired parameters. This array will be in BCD format and any integer or floating point parameters will have to be converted within the user supplied subroutine using CTOBIN. To convert an integer value in columns 19 through 24 (word 4) of CARD, the call to CTOBIN would be

CALL CTOBIN (1, CARD(4), FMT, 1, K)

Where FMT is the format specifying the type of conversion to be performed.

It may be specified with a DATA statement as follows:

DATA FMT/4H(I6)/

All tapes used by the user supplied subroutine must be assigned and released within the subroutine. All tapes should be rewound at the end of the VARY subroutine.

15 March 1972

2-38

TM-(L)-HU-033/006/00

By using one of the input parameters as a control switch, the user can set up his subroutine to perform multiple, independent functions.

Subroutine VARY is in LINK 1 of the IDAPS system. To avoid overflow of core storage, the length of VARY should be limited to 16,000₍₁₀₎ words.

2.31 MACRO Operator

The MACRO operator is used to group a series of operators such that each occurrence of the MACRO operator causes the entire series to be executed. This operator may be used when a repetitive type operation is called for.

Col.

1

MACRO (Any desired parameters)

This operator will initiate a call to subroutine MAC. The user must insert the necessary set of FORTRAN statements which call the desired operators in subroutine MAC. Subroutine MAC is in LINK 0 of the IDAPS system and therefore, its length should be held to a minimum (2000₍₁₀₎ words) to avoid overflow of core.

The SUBROUTINE statement in subroutine MAC will be as follows:

SUBROUTINE MAC (NL, NC, CARD)

NL - Number of lines from preceding SIZE operator
NC - Number of columns from preceding SIZE operator
CARD - An array containing the image of the MAC operator card which may contain any desired parameters. This array will be in BCD format and any integer or floating point parameters will have to be converted within the user supplied subroutine using CTOBIN. To convert an integer value in columns 19 through 24 (word 4) of CARD, the call to CTOBIN would be

CALL CTOBIN (1, CARD(4), FMT, 1, K)

Where FMT is the format specifying the type of conversion to be performed. It may be specified with a DATA statement as follows:

DATA FMT/4H(I6)/

By using one of the input parameters as a control switch, the user can set up his subroutine to perform multiple, independent functions.

2.32 STACK Operator

The STACK operator stacks multiple files on a single tape -- B5.

Col.

1



STACK

Normal tape assignments

A5 - Input

B5 - Output (Note that B5 is not a standard IDAPS tape unit. It is being used for this operator and in conjunction with MULTIPLE DISPLAY in order to avoid excessive tape handling.)

Data type:

Input - Integer

Output - Integer

Subroutines:

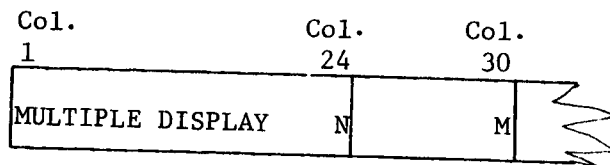
STKPIC

Restrictions:

- 1) When the output of STACK is used as input to MULTIPLE DISPLAY, all stacked images should be the same size.

2.33 MULTIPLE DISPLAY Operator

The MULTIPLE DISPLAY operator produces a display tape of 2 or more pictures



N is the number of pictures to be put on one output frame. If N is positive, all pictures will be taken from succeeding files of the input tape generated by operator STACK (see 2.32). If negative, program will pause prior to processing each input picture to allow operator to change tapes so that input may be taken from multiple tapes.

M is the number of pictures to be placed across the output frame. If positive, pictures will be separated by white space. If negative, pictures will be butted together.

Normal tape assignments:

B5 - Input (Note that B5 is not a standard IDAPS tape unit. It is being used for this operator and in conjunction with operator STACK in order to avoid excessive tape handling.)

A5 - Scratch

A6 - Output

Data type:

Input - Integer

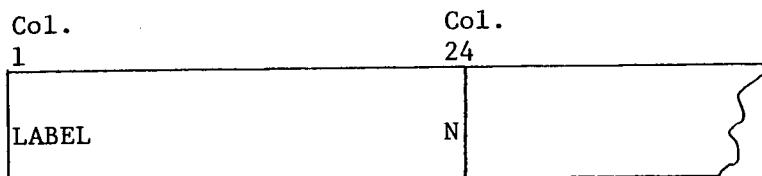
Output - Integer (display format)

Subroutine:

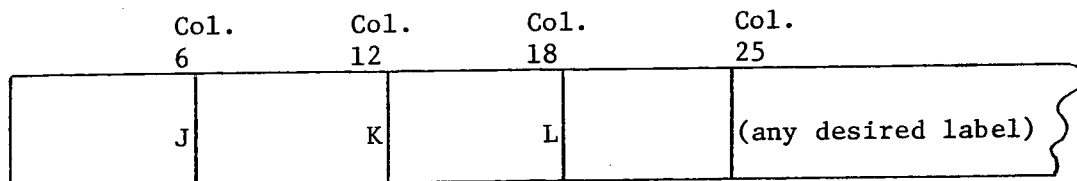
MULDSP

2.34 LABEL Operator

The LABEL operator inserts printed labels on a frame of data.



N is the number of label cards which follow the LABEL operator.



J is the number of characters in the label.

K is the starting line position on the frame.

L is the starting column position.

The label begins in column 25 and continues for J columns. If the label is longer than 48 characters, it may be continued on a second card beginning in column 1.

Each character is a 9 x 7 matrix with a 9 x 3 matrix of trailing blanks; therefore, each character occupies 9 lines and 10 columns. Line spacing must be controlled by the user -- a minimum of 5 lines between labels is recommended.

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Integer (display format)
Output - Integer (display format)

Subroutines:

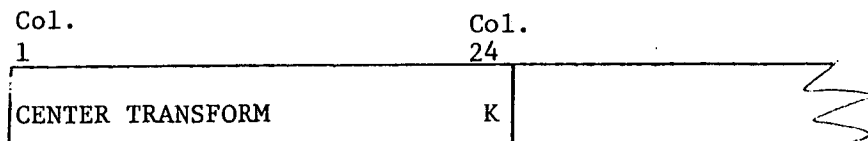
LABL
DSPLIN
UNPAC

Restrictions:

- 1) The maximum number of characters per line (including spaces) is 102.
- 2) Column 72 is the last column of a label card that will be interpreted.
- 3) The starting line position specified on each successive label card must be greater than the line position specification of the previous label card.
- 4) All labels should be limited to the following character set:
ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890+-(),/='* (space).

2.35 CENTER TRANSFORM Operator

The CENTER TRANSFORM operator multiplies an image by $(-1)^{i+j}$ where i is the row location and j is the column location of an image point. This has the effect of causing the Fourier transform of the image to be centered such that the d.c. term is located at the center of the transformed array.



K is the data type flag.

K = 1 - Integer

K ≠ 1 - Floating point

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Floating point or integer

Output - Floating point or integer

Subroutines:

CENTR

Note: To center an image transform, the CENTER TRANSFORM operator is applied before the FFT operator. To restore a picture from a centered transform, the CENTER TRANSFORM operator is applied after the IFFT operator.

2.36 SPLIT Operator

The SPLIT operator reads in complex data (real-imaginary or magnitude-phase) from one tape where it is stored as one line real (magnitude), one line imaginary (phase), etc., and outputs the data onto two tapes -- real (magnitude) on one and imaginary (phase) on the other. The routine also has the option of printing the data off-line as it is transferred.

Col. 1	Col. 24
SPLIT	NP

NP = 1 Print Real (Magnitude) values

NP = 2 Print Imaginary (Phase) values

NP = 3 Print Real and Imaginary (magnitude and phase) Values

NP = Other - No print

Normal tape assignments:

A5 - Input

A6 - Real (magnitude) output

A7 - Imaginary (phase) output

Data Type:

Input - Floating point

Output - Floating point

Subroutines:

SPLT

2.37 MERGE Operator

The MERGE operator reads in data from two input tapes and interleaves the complex parts (real/imaginary or magnitude/phase) line by line onto one tape.

Col.

1



MERGE

Normal tape assignments:

- A5 - Real (magnitude) input
- A6 - Imaginary (phase) input
- A7 - Interleaved output

Data type:

- Input - Floating point
- Output - Floating point

Subroutines:


MERG

2.38 M/P TO R/I Operator

The M/P TO R/I operator reads in a tape containing interleaved (line by line) magnitude/phase data, converts it to real and imaginary data, then writes out the real/imaginary data onto a tape interleaved line by line.

Col.

1



M/P TO R/I

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Floating point

Output - Floating point

Subroutines:


MPTORI

2.39 R/I TO M/P Operator

The R/I TO M/P operator reads in a tape containing interleaved (line by line) real/imaginary data, converts it to magnitude and phase data, then writes out the M/P data onto a tape interleaved line by line.

Col.

1



R/I TO M/P

Normal tape assignments:

A5 - Input

A6 - Output

Data type:

Input - Floating point

Output - Floating point

Subroutines:

RITOMP

2.40 LOG MAGNITUDE Operator

The LOG MAGNITUDE operator reads a magnitude/phase transform tape and takes the natural log of the magnitude values and uniformly quantizes the resulting spread into 64 gray levels. It also quantizes the phases to 60 gray levels such that gray level 60 equals $+180^\circ$, gray level 30 equals 0° , and gray level 0 equals -180° . The magnitude and phase gray levels are written on separate tapes.

Col.

1



LOG MAGNITUDE

Normal tape assignments:

A5 - Magnitude/Phase Interleaved Input

A6 - Magnitude Output

A7 - Phase Output

Data type:

Input - Floating point

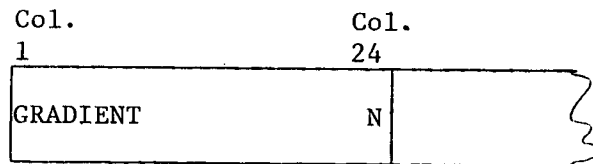
Output - Fixed point

Subroutines:

LOGMAG

2.41 GRADIENT Operator

The GRADIENT operator performs a position dependent alteration of the real and/or imaginary value of each element of a transformed picture.



N specifies the length of tables A, B, and C right justified to column 24. The values of the A, B, and C tables represent the coefficients of the expression $(A(M) + B(M)X + C(M)X^2)$ PIC (i,j) , where PIC (i,j) is the original picture value; $X = u + v$, where u is ω_x expressed in cycles/frame and v is ω_y expressed in cycles/frame. The index "M" ($M = \sqrt{u^2 + v^2}$) of the A, B, and C tables is determined by the radial distance from the center of the frame to the element. The tables are input immediately following the GRADIENT operator. For example, if N is 10; the A, B, and C may be input as follows:

Col.*

1

A = 3*0.0,2*1.,2.,3.,5.,8.,12.\$

B = 3*1.,.9,.7,.6,.5,.45,.42,.42\$

C = 5*1.,.4,.3,.2,.1,.05\$

Normal tape assignments:

A5 - Real or Imaginary Input

A6 - Output

Data type:

Input - Floating point

Output - Floating point

Subroutines:

PDAFNL


NOTE: Transform data should be split and centered prior to using this routine.

Use this routine on each (real, imaginary) half of data, then recombine.

*Continuation cards begin in column 2.

2.42 REWIND Operator

The REWIND operator rewinds a tape.

Col. 1	Col. 24	Col. 30	
REWIND	N	K	

N is the tape logical unit number.

K is a flag to unload the tape.

K = 1 - unload

K = other - do not unload

Normal tape assignments:

(None - must be assigned by user)

2.43 MATH Operator

The MATH operator performs the following point-by-point operations on two frames of data -- $A+B$, $A-B$, $A*B$, and A/B -- also $A+C$ and $A*C$ where C is a constant.

Col. 1	Col. 24	Col. 30	Col. 36
MATH	K	L	C

K is the operation flag

$K = 1$	-	$A+B$
$K = 2$	-	$A-B$
$K = 3$	-	$A*B$
$K = 4$	-	A/B
$K = 5$	-	$A+C$
$K = 6$	-	$A*C$

L is the data type flag

$L = 1$	-	integer (FORTRAN Format)
$L = 2$	-	Integer (display or scanner format)
$L = \text{other}$	-	floating point

C is a floating point constant to be used when $K = 5$ or 6 .

Normal Tape Assignments:

For $K = 1-4$

A5	-	Input A
A6	-	Input B
A7	-	Output

For $K = 5-6$

A5	-	Input
A6	-	Output

15 March 1972

2-53

TM-(L)-HU-033/006/00

Data Type:

(See L above)

Subroutines:

ARITH

UNPAC

2.44 OPERATOR SUMMARY

Operator	Ref.	NORMAL TAPE ASSIGNMENT			DATA TYPE	
		Input	Output	Scratch	Input	Output
AVERAGE	2.1	A5, A6...	$n^{th} + 1$		int.	int.
INTEGRATE	2.2	A5	A6		fl. pt.	fl. pt.
SCALE	2.3	A5	A6		fl. pt.	fl. pt.
INVARIANT ALTER	2.4	A5	A6		int.	int.
DEPENDENT ALTER	2.5	A5	A6		int.	int.
CONVOLVE (Spatial)	2.6	A5, A6	A7		int.	int.
DECONVOLVE	2.7	A5, A6	A7	A9,B6,B7,B9	fl. pt.	fl. pt.
SIZE	2.8					
COPY	2.9	Assigned by User			int./fl.pt.	int./fl.pt.
TRANSPOSE	2.10	A5	A6		int.	int.
EXPAND	2.11	A5	A6		int.	int.
CHANGE FORMAT	2.12	A5	A6		int.	int.
DISPLAY FORMAT	2.13	A5	A6		int.	int.
STOP	2.14					
RESOLUTION CHART	2.15		A5			int.
PRINT	2.16	A5			int./fl.pt	
PSF GENERATOR	2.17		A5			fl. pt.
EXTRACT	2.18	A5	A6		int.	int.
FORCON	2.19	A5, A6	A7	A9,B6,B7,B9	fl. pt.	fl. pt.
CORRELATE	2.20	A5, A6	A7	A9,B6,B7,B9	fl. pt.	fl. pt.
FFT	2.21	A5	A6	A7,A9	fl. pt.	fl. pt.
IFFT	2.22	A5	A6	A7,A9	fl. pt.	fl. pt.
FEATHER	2.23	A5	A6		int.	int.
FIXED TO FLOAT	2.24	A5	A6		int.	fl. pt.
FLOAT TO FIXED	2.25	A5	A6		fl. pt.	int.
ASSIGN	2.26					
RELEASE	2.27					
COMMENT	2.28					
PAUSE	2.29					
VARIABLE	2.30	Assigned by User				
MACRO	2.31	Assigned by User				

15 March 1972

2-55

TM-(L)-HU-033/006/00

Operator	Ref.	NORMAL TAPE ASSIGNMENT			DATA TYPE	
		Input	Output	Scratch	Input	Output
STACK	2.32	A5	B5		int.	int.
MULTIPLE DISPLAY	2.33	B5	A6	A5	int.	int.
LABEL	2.34	A5	A6		int.	int.
CENTER TRANSFORM	2.35	A5	A6		int./fl.pt.	int./fl.pt.
SPLIT	2.36	A5	A6,A7		fl. pt.	fl. pt.
MERGE	2.37	A5, A6	A7		fl. pt.	fl. pt.
M/P TO R/I	2.38	A5	A6		fl. pt.	fl. pt.
R/I TO M/P	2.39	A5	A6		fl. pt.	fl. pt.
LOG MAGNITUDE	2.40	A5	A6,A7		fl. pt.	int.
GRADIENT	2.41	A5	A6		fl. pt.	fl. pt.
REWIND	2.42					
MATH	2.43	A5, A6	A7		int./fl.pt.	int./fl.pt.

15 March 1972

3-1

TM-(L)-HU-033/006/00

SECTION 3. SOFTWARE

This section provides a description of the software modules of IDAPS.

IDAPS (Executive Control for IDAPS)

Purpose: IDAPS performs the following functions:

- 1) Reads and interprets IDAPS operators and their associated parameters
- 2) Assigns input, output and scratch tapes as required.
- 3) Calls the necessary subroutines
- 4) Releases the assigned tapes at the end of execution of each operator

Method: IDAPS utilizes the overlay feature of the IBLDR subsystem of IBSYS. The executive (IDAPS), data input routines (DATAIN, PACK, UNPACK, UNPAC) and the user supplied MAC occupy link 0. All other subroutines occupy subordinate links.

Storage: 3643₍₈₎ words

DATAIN (Data Input)

Purpose: DATAIN reads free field, tabular data from cards and stores it in an array.

Usage: CALL DATAIN (X, N, K)

X - Array into which data will be stored

N - Number of data values to be input

K - Data format

K = 0 - Integer

K = 1 - Floating point

Method: Data is input in the following general format:

A = 9., 8.1, 6.3, 4.2, 2.7, 1.6, .8, .09, 12*0.\$

DATAIN looks for the equal sign and then isolates each number separated by commas until a \$ is encountered at which point the input is terminated.

Multiple values may be input by using an asterisk as above (12*0.) to indicate that 12 zeros are to be input following the .09.

Integer values are input without decimal points.

Continuation cards are required if data cannot be contained in columns 1 - 78 of the first card. Continuation cards must start in column 2 with column 1 blank.

Storage: 703₆₈ words

Required by: INVARIANT ALTER
DEPENDENT ALTER
PSF GENERATOR
GRADIENT
FEATHER

15 March 1972

3-4

TM-(L)-HU-033/006/00

PACK (Pack 6 characters per word)

Purpose: PACK transfers "n" characters which are stored as the least significant 6 bits of a word into an array where the characters are packed 6 per word.

Usage: CALL PACK (A, N, B)
A - Array of characters to be packed
N - Number of characters
B - Array into which characters will be packed

Storage: 31₍₈₎ words

Required by: INVARIANT ALTER
DEPENDENT ALTER
PSF GENERATOR
GRADIENT
FEATHER

15 March 1972

3-5

TM-(L)-HU-033/006/00

UNPACK (Unpack 1 character from packed array)

Purpose: UNPACK is a function which extracts the j^{th} character from a packed array.

Usage: B = UNPACK (A,J)
A - array of packed data
J - index of character to be unpacked
B - desired character right justified with leading blanks

Storage: 66_8 words

Required by: INVARIANT ALTER
DEPENDENT ALTER
PSF GENERATOR
GRADIENT
FEATHER

UNPAC (Unpack 1 character from packed array)

Purpose: UNPAC is a function which extracts the j^{th} character from a packed array.

Usage: $B = \text{UNPAC}(A, J)$
A - array of packed data
J - index of character to be unpacked
B - desired character right justified with leading zeros.

Storage: $65_{(8)}$ words

Required by: DSPLIN

MAC

Purpose: MAC provides the capability to perform repetitive operations without duplicating the IDAPS operators. The user supplies the sequence of operations desired by inserting the necessary FORTRAN code into the basic MAC subroutine framework.

Usage: CALL MAC (NL, NC, CD)
NL - Number of lines
NC - Number of columns
CD - Operator card image in BCD format

Method: Subroutine MAC is in link 0 of IDAPS to permit direct calling of the link 1 subroutines. Any set of FORTRAN code may be inserted into the basic MAC subroutine framework.

Restrictions: (1) The total length of MAC, including data, should be limited to less than 2000 words.

Required by: MACRO

ASSGN (Assign and Release Tapes)

Purpose: ASSGN assigns all standard input, output and scratch tapes for each operator within IDAPS. FREE, an entry within ASSGN, releases assigned tapes.

Usage: CALL ASSGN (I, J)

I - Control flag

I = 0 Output tape

I = 1, 2, ... Input tape for operator I

I = -1 Scratch Tape

J - Logical unit number assigned by subroutine ASSGN

CALL FREE (J)

J - Logical tape unit to release

Method: Tapes are assigned in the following order:

A5(8), A6(10), A7(12), A9(14), B6(11), B7(13), B9(15)

Restrictions: (1) A maximum of 7 tapes may be assigned at any one time.

Storage: 603₈ words

Required by: All IDAPS operators requiring tapes.

15 March 1972

3-9

TM-(L)-HU-033/006/00

FREE (Free Tape)

(See ASSGN Subroutine write-up)

VARY

Purpose: VARY provides the capability of inserting temporary, one-shot subroutines in IDAPS without modifying the system driver.

Usage: CALL VARY (NL, NC, CD)

NL - Number of lines
NC - Number of columns
CD - Parameter array in BCD format
(Contains image of VARIABLE operator input card)

Method: Subroutine VARY is in link 1 of IDAPS. Any set of FOTRAN code may be inserted into the basic framework of the VARY subroutine. By using one of the parameters on the input card as a switch, the user can set up subroutine VARY to perform multiple, independent functions.

Restrictions: (1) The total length of VARY, including data, should be limited to less than 16,000 words.

Required by: VARIABLE

CONVRT (Scanner to FORTRAN Tape Format Conversion)

Purpose: CONVRT converts from packed format (6 elements per word) to a single element per word format.

Usage: CALL CONVRT (IN, IO, KL, KC, NL, NC, KF)

IN - Input tape logical unit number
 IO - Output tape logical unit number
 KL - Starting line number of input frame
 KC - Starting column number of input frame
 NL - Number of lines per output frame
 NC - Number of columns per output frame
 KF - File number of input frame

Method: REDTPR (General Purpose Tape Read Routine) is used to read the input tape. FLD (Bit Transfer Routine) is used to unpack the data. Each line of the frame is output in FORTRAN format.

Restrictions: (1) The maximum size of the input frame is limited to 1536 columns (256 words) since this is the size of the system I/O buffers. If larger frames are required, the I/O buffer size must be changed by altering the buffer size. This may be done by replacing the system .UNnn. routine. For example, if the input on unit 8 is 2048 elements (342 words), .UN08. would be changed as follows:

Col.	Col.	Col.
1	8	16
\$IBMAP	UN8	
	ENTRY	.UN08.
.UN08.	PZE	UNIT08
UNIT08	FILE	,CK1, READY, INOUT, BLK=342, BIN, NOLIST
	END	

The absolute maximum frame size is 2048 by 2048.

15 March 1972

3-12

TM-(L)-HU-033/006/00

Storage: 5026 (8 words

Required by: CHANGE FORMAT

MLTAVS (Multi-Copy Average)

Purpose: MLTAVS computes the average of up to five frames of data.

Usage: CALL MLTAVS (NF, NL, NC, KT, IO)

NF - Number of frames to be averaged

NL - Number of lines per frame

NC - Number of columns per frame

KT - Location of table specifying logical unit number of each
input tape

IO - Output tape logical unit number

Method: For corresponding picture elements of the frames of data to be averaged, the mean gray scale value is calculated. The standard deviation is computed as

$$s = \sqrt{\frac{\sum_{i=1}^n X_i^2 - \frac{(\sum_{i=1}^n X_i)^2}{n}}{n-1}}$$

Those gray scale values outside 1 standard deviation are eliminated and a new mean value is calculated using only the gray scale values within 1 standard deviation of the original mean.

Restrictions: (1) A maximum of 5 frames of data may be averaged.
(2) The maximum size of the frames to be averaged is
2048 by 2048.

Storage: 30443 (8 words)

Required by: AVERAGE

SPINT (Spatial Integration)

Purpose: SPINT performs a spatial integration of a frame of data.

Usage: CALL SPINT (IN, IO, NL, NC, LM, AJ)

IN - Input tape logical unit number

IO - Output tape logical unit number

NL - Number of lines per frame

NC - Number of columns per frame

LM - Allowable variance within which an element will not be adjusted

AJ - Adjustment factor (stated as a percentage) by which an element outside the allowable variance of the average of its neighbors will be adjusted.

Method: Each element of the frame of data within a 1 element border is surrounded by 8 immediate neighbors; i.e.,

N	N	N
N	E	N
N	N	N

The mean (\bar{x}) of the 8 neighboring elements is computed for each element within the frame. If the center element (E) is outside a specified limit of the mean of its neighbors, it is replaced by adjusting it as follows:

$$\text{New E} = E - [(E - \bar{x}) \cdot \text{AJ}]$$

For example, if $E=20$, $\bar{x}=28$, $\text{LM}=2$, and $\text{AJ}=.8$ (80%):

$$\begin{aligned}\text{New E} &= 20 - [(20 - 28) \cdot .8] \\ &= 20 - (-6.4) \\ &= 26.4 \approx 26\end{aligned}$$

15 March 1972

3-15

TM-(L)-HU-033/006/00

Restrictions: (1) The maximum frame size is 2048 by 2048.
(2) The outer border of the frame is unaltered,

Storage: 20431 (8 words

Required by: INTEGRATE

XPAND (Frame Expansion)

Purpose: XPAND performs an expansion or an extraction of a selected portion of a frame of data.

Usage: CALL XPAND (IN, IO, LL, LC, NE, NP, NL, NC)

IN - Input tape logical unit number
IO - Output tape logical unit number
LL - Position of left upper line
LC - Position of left upper column
NE - Size of square area to be expanded or extracted
NP - Expansion factor
NL - Number of lines in original frame
NC - Number of columns in original frame

Method: Each line of the original picture is read from the input tape and each element within the area to be expanded is duplicated NP times. Each line of data is then output NP times on the output tape. To extract a selected portion of a frame of data rather than expand, NP should be set to 1.

Restrictions: (1) NE x NP may not exceed 2048.

Storage: 10362₍₈₎ words

Required by: EXPAND, EXTRACT

CNVOL (Numeric Convolve)

Purpose: CNVOL performs a numeric convolution of an input image frame with an input PSF.

Usage: CALL CNVOL (JIN, JOUT, JFIN, NLIN, NCLM, KSIZ)
JIN - Image input tape logical unit number
JOUT - Output tape logical unit number
JFIN - PSF input tape logical unit number
NLIN - Number of lines per image input frame
NCLM - Number of columns per image input frame
KSIZ - Size of one side of PSF array

Method: Each of the picture elements of the input picture array is multiplied by all of the matrix points of the PSF array and the resulting arrays are added together where they spatially overlap. Mathematically this is expressed as

$$g(I,J) = \frac{1}{N} \sum_{x=-a}^{+a} \sum_{y=-b}^{+b} e(I-y, J-x)f(x,y)$$

where $I=1,2,\dots,NLIN$ and $J=1,2,\dots,NCLM$ and $a=b=(KSIZ-1)/2$.
N is a normalization factor which is the volume under the PSF surface except in those regions where the PSF overlaps the edge of the input picture. In those cases N is only that volume under the PSF surface which overlaps the input picture.

Restrictions: (1) NCLM must not exceed 128 picture elements
(2) KSIZ must not exceed 63 and must be an odd number

Storage: 23035₍₈₎ words

Required by: CONVOLVE

SHADE (Position Invariant Gray Scale Alteration)

Purpose: SHADE modifies the gray scale values of an image using the original gray value as an index to a table of modified gray values.

Usage: CALL SHADE (IN, IO, NL, NC)

IN - Input tape logical unit number

IO - Output tape logical unit number

NL - Number of lines per frame

NC - Number of columns per frame

The modification table is input by SHADE as follows:

Col.

1

K=(desired set of 64 values)\$

Storage: 4305₍₈₎ words

Required by: INVARIANT ALTER

PDGSA (Position Dependent Gray Scale Alteration)

Purpose: PDGSA modifies each element of a frame of data as a function of its position within the frame.

Usage: CALL PDGSA (NL, NC, IN, IO, NP)

NL - Number of lines per frame

NC - Number of columns per frame

IN - Input tape logical unit number

IO - Output tape logical unit number

NP - Number of points in the modification tables

The modification tables are input by PDGSA as follows:

Col.*

1

A=(up to 100 values)\$

B=(up to 100 values)\$

The values of tables A and B represent the coefficients of the linear expression $A + BX$.

Method: The radial distance of an element from the center of the frame is computed and is used as the index (m) to tables A and B in calculating a modified value as

$$\text{Modified value} = A_m + B_m \times \text{original value}$$

The modified value replaces the original value within the frame.

15 March 1972

3-20

TM-(L)-HU-033/006/00

Restrictions: (1) Tables A and B may not exceed 100 values.
(2) All alterations are symmetrical about the center point in the frame of data.

Storage: 10743₍₈₎ words

Required by: DEPENDENT ALTER

DISPLAY (Convert to Display Format)

Purpose: DISPLAY converts a single frame of data from FORTRAN to display format.

Usage: CALL DISPLAY (IN, IO, NL, NC, NG)
IN - Input tape logical unit number
IO - Output tape logical unit number
NL - Number of lines per frame
NC - Number of columns per frame
NG - Grid flag
 G = 1 - overlay grid
 G = other - no grid

Method: PUPINS (Pack and Unpack Integers) is used to pack a line of data 6 bits per word and WRITER (General Purpose Tape Write Routine) is used to write the line in the format of the DICOMED display and the Optronix Photowrite.

Restrictions: (1) The maximum frame size is 1024 x 1024.

Storage: 5277₍₈₎ words

Required by: DISPLAY FORMAT

TRNPS (Transpose)

Purpose: TRNPS transposes a frame of integer data replacing rows by columns.

Usage: CALL TRNPS (IN, IO, NL, NC)
IN - Input tape logical unit number
IO - Output tape logical unit number
NL - Number of lines
NC - Number of columns

Method: In matrix notation
 $OUTPUT = INPUT^T$

Restrictions: (1) The maximum frame size is 256 x 256.
(2) Both input and output are in integer format.

Storage: 31774_8 words

Required by: TRANSPOSE

15 March 1972

3-23

TM-(L)-HU-033/006/00

RSLGEN (Image Resolution Chart Generator)

Purpose: RSLGEN produces a two-dimensional resolution chart consisting of alternating groups of black and white bars with spatial frequencies which range from one cycle/line to one-half cycle per column.

Usage: CALL RSLGEN (JOUT, SIZE)

JOUT - Output tape logical unit number

SIZE - Desired size of the square output array (up to 2048).

Storage: 4345₈ words.

Required by: RESOLUTION CHART

PSFGEN (Symmetrical Point Spread Function Generator)

Purpose: PSFGEN accepts a list of numbers which define a point spread function half-profile and creates a three-dimensional PSF array in floating point.

Usage: CALL PSFGEN (JOUT, KSIZ)

JOUT - Output tape logical unit number

KSIZ - Desired size of the square output array (must be larger than the number of points in the PSF input table and less than 100)

The PSF to be generated is input by a table as follows:

Col

1

CURVE = (desired set of up to 50 floating point values between 0. and 63.)\$

Method: The three-dimensional PSF is generated by rotating the input curve about its left hand axis. The distance from the center is calculated for each point and a gray value is calculated by interpolating between the input CURVE points using library routine LAGRNG.

Storage: 24152₍₈₎ words

Required by: PSF GENERATOR

HRDCPY (Hardcopy Printout of Picture Arrays)

Purpose: HRDCPY formats a frame of picture data and prepares an output tape for off-line printout or prints directly online in either fixed or floating point, sixteen levels of gray, or a map of values above a specified threshold.

Usage: CALL HRDCPY (JIN, ITOP, JLFT, KLMNS, LINES, NL, NC, NO)

JIN - Input tape logical unit number

ITOP - Starting line number

JLFT - Starting column number

KLMNS - Number of columns to be printed

LINES - Number of lines to be printed

NL - Number of lines in the input image array

NC - Number of columns in the input image array

NO - Control flag which if --

NO = -4 Print gray level values on-line in floating point.

NO = -3 Print gray level values off-line in floating point.

NO = -2 Prints gray level values on-line in integer form.

NO = -1 Prints gray level values off-line in integer form.

NO = 0 to 63 Prints blanks for values less than or equal to NO and X for values greater than NO.

NO = 64 Prints 16 levels of gray using overstrike method.

Method: When printing gray level output, HRDCPY divides each gray level value (integer values 0 to 63) by four, adds one and then prints up to nine overstrikes of selected characters to produce one of sixteen different gray levels.

Storage: 15170₍₈₎ words.

Required by: PRINT

CONV (Convolution Driver)

Purpose: CONV provides the driver for the following two-dimensional Fourier transform operations.

Convolve

Deconvolve

Correlate

Transform

Inverse transform

Usage: The subroutine has five entry points

Convolution

```
CALL CONV (IP, MSIZE, IN, NSIZE, IO, NTSIZE, KFLAG,  
           I1, I2, I3, I4)
```

Deconvolution

```
CALL DCONV (IP, MSIZE, IN, NSIZE, IO, NTSIZE, RMAG,  
            PHA, KFLAG, IT, TX, TN, I1, I2, I3, I4)
```

Correlation

```
CALL CRLATE (IP, MSIZE, IN, NSIZE, IO, NTSIZE, KFLAG,  
             I1, I2, I3, I4)
```

Transform

```
CALL XFORM (IP, MSIZE, IO, NTSIZE, I1, I2)
```

Inverse transformation

```
CALL IXFORM (IP, MSIZE, IO, NTSIZE, IT, TX, TN, I1, I2)
```

IP - Point spread function tape logical unit number

MSIZE - Size of point spread function

IN - Input image tape logical unit number

NSIZE - Size of input image

IO - Output image tape logical unit number

NTSIZE - Size of output image

RMAG - Limit on magnitude amplification - see SPRED2

PHA - Limit on phase amplification - see SPRED2

KFLAG - Point spread function control flag

KFLAG = 1 - Point spread function is being
input in frequency domain.

≠ 1 - Point spread function is being
input in spatial domain.

IT - Truncation flag

= 1 - No truncation

= 2 - Truncate outside TX and TN

= other - Truncate outside of 0. and 63.

TX - Maximum truncation limit

TN - Minimum truncation limit

I1 - Scratch tape

I2 - Scratch tape

I3 - Scratch tape

I4 - Scratch tape

Notes: If the point spread function is input in the spatial domain
(KFLAG ≠ 1), an extra call is made to FRX2V to transform and
normalize the PSF.

Restrictions: The point spread function and input image arrays must be square.

Storage: 1122₍₈₎ words

Required by: FFT, IFFT, CORRELATE, FORCON, DECONVOLVE

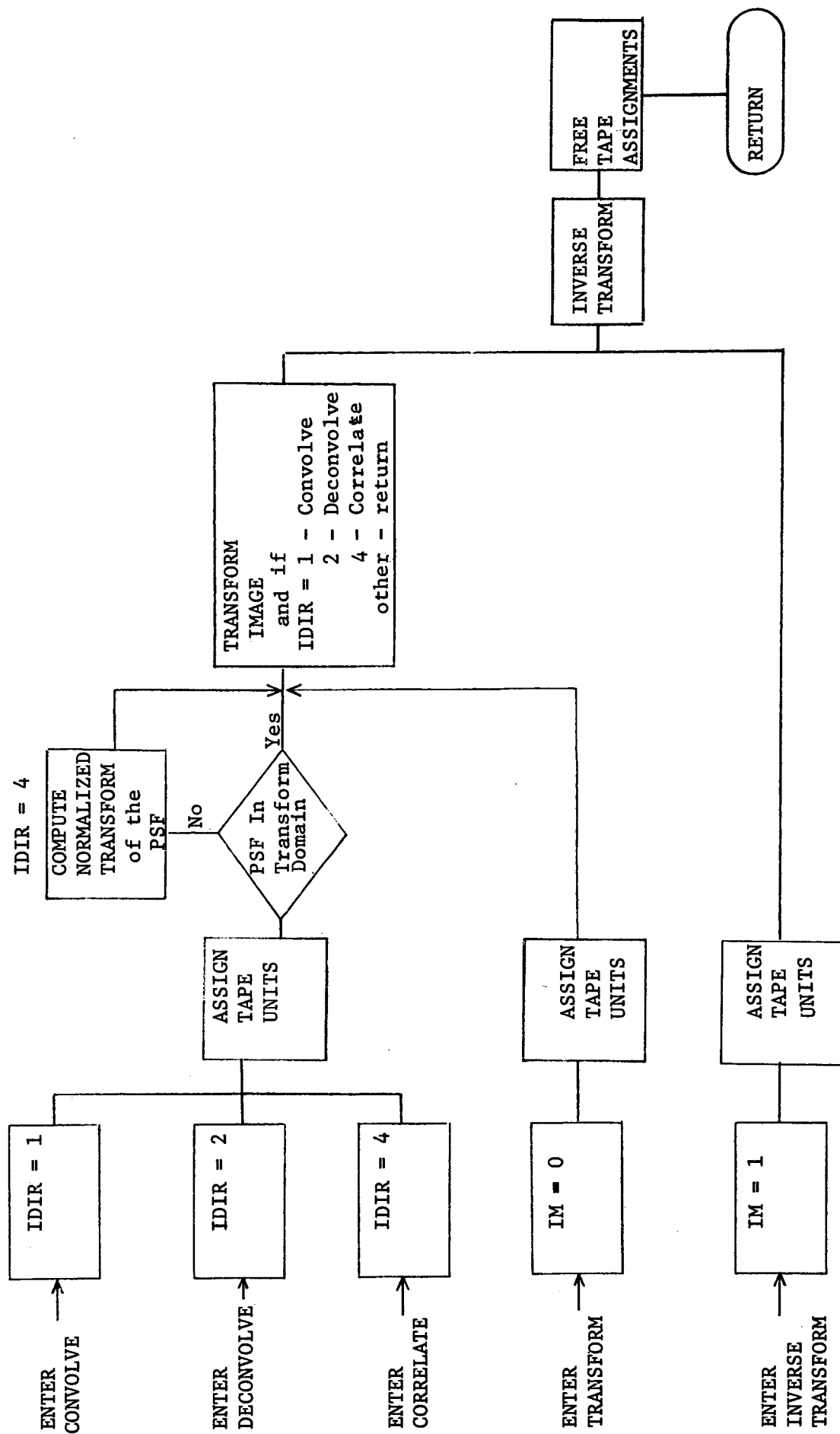


Figure 4 - Flow of Subroutine CONV

15 March 1972

3-29

TM-(L)-HU-033/006/00

DCONV (DECONVOLVE)

(See CONV Subroutine write-up)

15 March 1972

3-30

TM-(L)-HU-033/006/00

CRLATE (Correlate Driver)

(See CONV Subroutine write-up)

15 March 1972

3-31

TM-(L)-HU-033/006/00

XFORM (Fourier Transform)

(See CONV Subroutine write-up)

15 March 1972

3-32

TM-(L)-HU-033/006/00

IXFORM (Inverse Fourier Transform)

(See CONV Subroutine write-up)

FRX2V (TWO-DIMENSIONAL FAST FOURIER PROCESS DRIVER)

Purpose: FRX2V performs a two-dimensional Fourier transformation by calling the fast Fourier subroutine FRXFM first by rows then by columns. FRX2V also serves as the main processor for the convolution, deconvolution, correlation, transform, normalized transform, and inverse transform processes.

Usage: Call FRX2V (NSIZE, NPSIZE, IN, IW, IØ, IM, IAPPLY, IP, II, RMAG, PHA, ITF, TMAX, TMIN)

NSIZE - Input image dimension

NPSIZE - Output image dimension

IN - Input image tape logical unit number

IW - Scratch tape unit number

IØ - Output image tape logical unit number

IM - Forward/Inverse transformation flag
= 0 Forward transformation
= 1 Inverse

IAPPLY - Process control flag
= 1 Convolve
= 2 Deconvolve
= 3 Transform and normalize
= 4 Correlate
= other Transform

IP - Transformed point spread function tape unit number

II - Scratch tape unit number

RMHG - Limit on magnitude amplification -- see SPRED2

PHA - Limit on phase amplification -- see SPRED2

ITF - Truncation flag
= 1 No truncation
= 2 Truncate values outside of TMIN and TMAX
= other truncate values outside of 0. and 63.

TMAX - Maximum truncation limit

TMIN - Minimum truncation limit

Method:

A two-dimensional Fourier transformation is achieved by first performing a one-dimensional transformation of each of the rows of the original array and then performing a one-dimensional transformation of each of the columns of the array resulting from the first set of transformations. To better understand this procedure consider the continuous two-dimensional transform of equation (1)

$$h(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(X,Y) e^{-i2\pi ux} e^{-i2\pi vy} dx dy \quad (1)$$

If an interval T is divided into N equal parts both horizontally and vertically such that $N\Delta X = N\Delta Y = T$, then the maximum frequency $F = \frac{1}{\Delta X} = \frac{1}{\Delta Y}$. The incremental frequencies are then

$\Delta U = \Delta V = F/N$ so that $\Delta U \Delta X = \Delta V \Delta Y = \frac{1}{N}$. Using the indices j, k, ℓ, m equation (1) in the discrete case becomes

$$h(\ell \Delta u, m \Delta u) = \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} H(j \Delta x, k \Delta y) e^{\frac{-2\pi i}{N} j \ell} e^{\frac{-2\pi i}{N} k m} \quad (2)$$

Let

$$TR(j, m) = \sum_{k=0}^{N-1} H(j \Delta x, k \Delta y) e^{\frac{-2\pi i}{N} k m} \quad (3)$$

then

$$h(\ell \Delta u, m \Delta u) = \sum_{j=0}^{N-1} TR(j, m) e^{\frac{-2\pi i}{N} j \ell} \quad (4)$$

but a one-dimensional discrete Fourier transform over an interval T , sampled N times ($N\Delta X = T$ and $F = \frac{1}{\Delta X}$, $\Delta U = F/N$) is given by

$$T(M \Delta X) = \sum_{k=0}^{N-1} V(k) e^{\frac{-2\pi i}{N} k m}$$

Therefore equation (3) is just N ordinary one-dimensional transforms, one for each row of the picture, and equation (4) is N ordinary one-dimensional transforms, one for each of the columns resulting from the N applications of equation 3.

Image Buffering:

If the input picture(s) and/or point spread function, have dimensions less than the smallest power of two greater than NPSIZE they are buffered with zeros equally on all sides up to this power. Indeed for convolution, deconvolution, or correlation this is necessary in order to prevent image overlapping due to the cyclic nature of the fast Fourier transform. In general if M is the size of the input picture and L is the size of the point spread function, the NPSIZE should be at least the smallest power of two greater than $M + L$.

Point Spread Function Transformation:

If the PSF is input in the spatial domain, an extra call to FRX2V is made by CONV to transform and normalize the PSF (IAPPLY = 3). The normalization consists of summing the gray levels of all elements of the PSF and dividing each frequency element (both real and imaginary) by this amount. This procedure insures that a unit amount of light will be dispensed for each unit present in the input picture.

Image Centering:

The transform of the PSF (real and imaginary) is also multiplied by $(-1)^{i+j}$, $i=1, 2, \dots, \text{NPSIZE}$, $j=1, 2, \dots, \text{NPSIZE}$, for convolution and deconvolution so that the output picture will be centered in the frame.

Clipping:

Due to improper amplification of certain frequency terms due to noise, improper PSF, etc., the gray levels obtained from deconvolution may extend over a large dynamic range. The subroutine will determine the maximum and minimum values encountered and will print these on line. If the 64 levels

were divided over this large range one would obtain an output picture with little contrast. A much improved picture will result if the output is clipped both above and below, with the remaining levels quantized to 64 gray levels. Presently the subroutine clips at 0. and 63., that is, all values below zero are set to zero, and all values above 63 are set to 63. The remaining range is truncated to 64 gray levels.

Data Storage: In order to reduce tape movements on the IBM 7094 computer FRX2V has been written to process two-dimensional image arrays in blocks and to store (on tape) the transformed data in a form most suitable to minimize tape movement.

<u>Picture Size</u>	<u>No. of Blocks</u>	<u>No. of Rows of Block</u>	<u>No. of Columns of Block</u>
16 x 16	1	16	16
32 x 32	1	32	32
64 x 64	1	64	64
128 x 128	4	32	128
256 x 256	16	16	256
512 x 512	64	8	512

Tape storage format between 1-dimensional transform calls is:

16 x 16

(1,1)(1,2)...Real...(16,15)(16,16)	(1,1)(1,2)...Imaginary...(16,15)(16,16)
256	256

32 x 32

(1,1)(1,2)...Real...(32,31)(32,32)	(1,1)(1,2)...Imaginary...(32,31)(32,32)
1024	1024

64 x 64

(1,1)(1,2)...Real...(64,63)(64,64)	(1,1)(1,2)...Imaginary...(64,63)(64,64)
4096	4096

The subroutine calls the subroutine SPRED1 to achieve convolution, SPRED2 to achieve deconvolution, and SPRED3 to achieve correlation in the frequency domain.

Tape Requirements:

Convolution*	7 units
Deconvolution*	7 units
Correlation	7 units
Forward Transform	5 units
Inverse Transform	5 units

*Only 6 units are required if the transform of the PSF is input to these routines.

- Restrictions:
- (1) The image size must be greater than or equal to 16 by 16 and must be less than or equal to 512 by 512. This should be considered the maximum output image size.
 - (2) The output image dimension must be square and must be greater than or equal to NSIZE.
 - (3) The inverse operator assumes the input image is square.

Storage: 30000₍₈₎ words

Required by: FFT, IFFT, CORRELATE, FORCON, DECONVOLVE

Figure 5 presents the data storage layout for a 128 by 128 frame of data. Since the whole picture will not fit into core at the same time, the frame is divided into four sections and each section processed in turn. Frames of data which are 256 by 256, and 512 by 512 are similarly handled to minimize tape transfer.

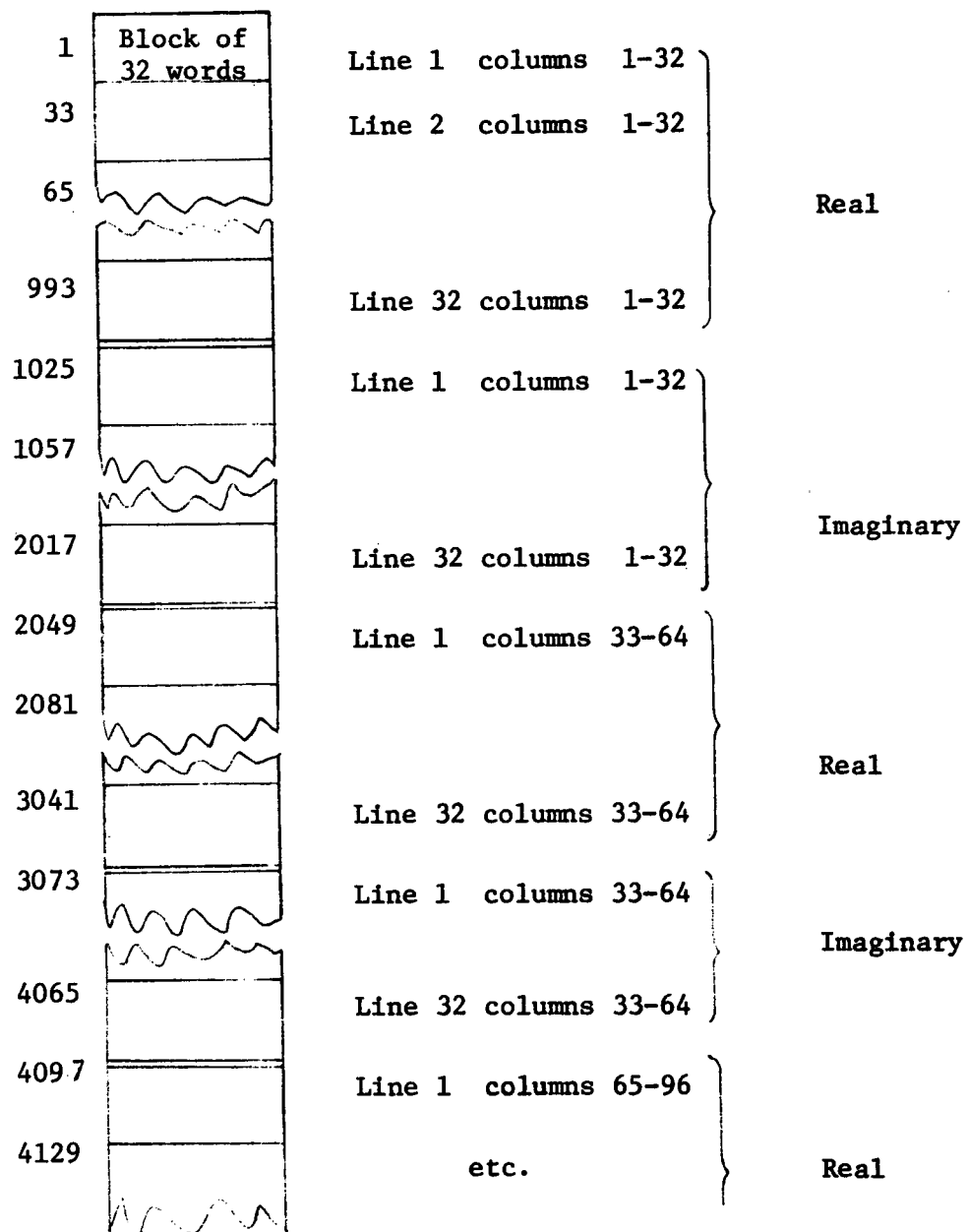


Figure 5 - 128 x 128 Data Storage Layout

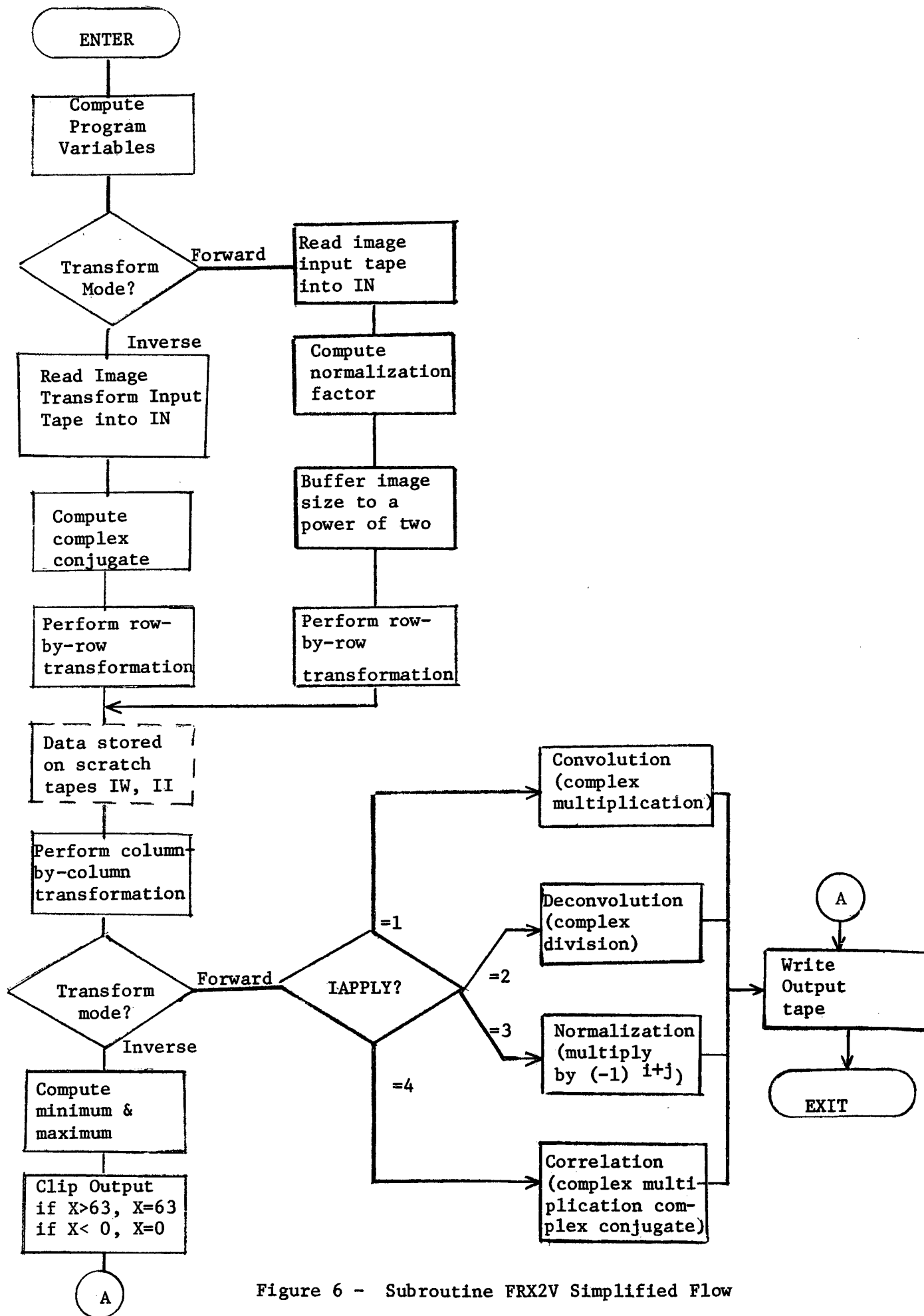


Figure 6 - Subroutine FRX2V Simplified Flow

FRXFM (Fast Fourier Transform Routine)

Purpose: To compute the fast Fourier transform of a complex vector using a Radix 4 plus 2 algorithm.

Usage: Call FRXFM (N2POW, X, Y).

N2POW -- If N is the number of elements in the complex array then $N2POW = \log_2 N$.

X -- Array containing the real part of the complex vector

Y -- Array containing the imaginary part of the complex vector.

Notes: The array need be dimensioned only in the calling routine. If a real array is to be transformed, set the Y array elements to zero and proceed as usual. The transform is performed in place, therefore the input data is destroyed as the transform is computed.

Method: The routine is a modified implementation of a procedure originally developed by Cooley and Tukey and later improved by Gentleman and Sande.

If N is the size of the vector to be transformed, then the equations describing discrete Fourier transformations are given by:

$$X(j) = \sum_{k=0}^{N-1} A(k) \exp \left(\frac{i2\pi jk}{N} \right) \quad j=0,1,\dots,N-1 \quad (1)$$

$$A(k) = \sum_{j=0}^{N-1} X(j) \exp \left(-\frac{i2\pi jk}{N} \right) \quad k=0,1,\dots,N-1 \quad (2)$$

If we let $W = \exp \left(\frac{i2\pi}{N} \right)$ and $W^\alpha = W^{nk}$, then equation (1) can be written in matrix form as

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & \dots & W^0 \\ W^0 & W^1 & & W^{N-1} \\ \vdots & \vdots & & \vdots \\ W^0 & W^{N-1} & \dots & X^{(N-1)^2} \end{bmatrix} \begin{bmatrix} A(0) \\ A(1) \\ \vdots \\ A(N-1) \end{bmatrix} \quad (3)$$

or

$$X = T A \quad (4)$$

In general, equation(4) requires N^2 multiplications and divisions. If, however, N has M factors r_1, r_2, \dots, r_m such that $N = \prod_{i=1}^m r_i$, then T can be decomposed into the product

of m elementary transformations, followed by a permutation P of the result:

$$T = P T_m T_{m-1} \dots T_1 \quad (5)$$

Each step T_j is in turn composed of N/r_j transforms of dimension r_j . Since the number of arithmetic operations for a transform of dimension r_j is of the order r_j^2 , the total number of operations for the transformation T is less than $N \sum_{j=1}^m r_j = mNr$

but $N=r^m$ or $\log_r N=m$, therefore computation requires $r_N \log_r N$ arithmetic operations compared with the N^2 operations required by a standard transform. Also by reduction of arithmetic operations, accuracy is improved.

The recursive equations needed to implement the fast Fourier transformation routine for arbitrary length records ($r=2^\alpha$ for all integer α) were derived by G. D. Bergland and are shown below.

$$A_p(j_0, j_1, \dots, j_{p-1}, k_{n-p-1}, \dots, k_0)$$

$$= \sum_{k_{n-p}}^{r_p-1} A_{p-1}(j_0, j_1, \dots, j_{p-2}, k_{m-p}, \dots, k_0) \cdot$$

$$W_N [j_{p-1}(r_1 r_2 \dots r_{p-1}) + \dots + j_0] k_{m-p}(r_{p+1} \dots r_m) \quad (6)$$

The last entry thus calculated will yield the Fourier sum as

$$X(j_{m-1}, \dots, j_0) = A_m(j_0, \dots, j_{m-1}) \quad (7)$$

in such an order that the index of an X must have its binary bits put in reverse order to yield its index in the array A_m .

In some cases the computation required to evaluate equation (6) can be reduced by regrouping the equation as

$$\hat{A}_p(j_0, j_1, \dots, j_{p-1}, k_{m-p-1}, \dots, k_0)$$

$$= \left[\sum_{k_{n-p}}^{r_p-1} \hat{A}_{p-1}(j_0, j_1, \dots, j_{p-2}, k_{m-p}, \dots, k_0) W_{r_p}^{j_{p-1} k_{n-p}} \right] \cdot$$

$$W_N^{j_{p-1}(k_{n-p-1}(r_{p+2} \dots r_m) + \dots + k_1 k_n + r_0)(r_1 r_2 \dots r_{p-1})} \quad (8)$$

The bracketed term represents a set of r_p -point Fourier transforms and the complex exponential weights outside the brackets reference each set of results to a common time origin.

The term

$$W_{r_p} = W_N^{N/r_p} = e^{2\pi i/r_p}$$

forms the basis for the complex exponential weights required in evaluating each r_p -point transform, and j_{p-1} , k_{m-p} are the indices of the transform.

If the r_1, r_2, \dots, r_m are all equal to some integer, say α , then the transformation described by the resulting recursive equations is referred to as a Radix α transformation. Cooley and Tukey stated in their original paper that the FFT algorithm is formally most efficient when the number of samples in a record is a power of 3 (i.e. $r_j = 3$) but such an algorithm is only slightly more efficient than a Radix 2 or a Radix 4 routine. Later it was recognized that the symmetries of the sine and cosine weighting functions made the Radix 4 algorithm more efficient than either the Radix 2 or Radix 3 algorithm. Radix 4 algorithms, however, suffered from the severe limitation on acceptable record sizes. To overcome this difficulty Gentleman and Sande developed an algorithm which performs as many iterations of the transform as possible with a Radix 4 routine, and then, if required, performs the last iteration with a Radix 2 routine. A modified version of Gentleman and Sande's algorithm is implemented as the subroutine FRXFM.

If N is the size of the array such that $N = 2^{N2POW}$, then $N2POW$ is divided by two (integer division) to compute the required number of transformations of dimension four ($N4POW$).

Restrictions: The dimension (N) of the input array must be a power of two. If it is not, zeros may be used to buffer it up to a length which is a power of two.

15 March 1972

3-44

TM-(L)-HU-033/006/00

Tape Requirements:

None

Storage: 1224₍₈₎ words

Required by: FFT, IFFT, CORRELATE, FORCON, DECONVOLVE

SPRED1

Purpose: SPRED1 is designed to perform multiplication of two complex vectors.

Usage: CALL SPRED1 (N, IP, X, Y, GR, GI)

N -- Size of the arrays
IP -- Tape logical unit containing PSF transform array
X -- Real part of image transform array
Y -- Imaginary part of image transform array
GR -- Real part of PSF transform array
GI -- Imaginary part of PSF transform array

Method: Let $I = \text{image} = X + j Y$
 $P = \text{PSF} = GR + j GI$
 $O_R = \text{real part of resulting image and,}$
 $O_I = \text{imaginary part of resulting image.}$
then $(I)(P) = (X + jY) (GR + j GI) =$
 $(X)(GR) - (Y)(GI) + j [(Y)(GR) + (X) (GI)]$
 $O_R = (X)(GR) - (Y)(GI)$
 $O_I = (Y)(GR) + (X)(GI)$

Restrictions: The arrays must be the same size and less than or equal to size 1024.

Array number two is read from tape and must have been written in blocks of N real, N imaginary, N real, N imaginary, etc.

Storage: 166₍₈₎

Required by: FORCON

SPRED2

Purpose: SPRED2 is designed to perform the division of two complex vectors while allowing limiting of the amount of magnitude change or phase change of the numerator resulting from this operation.

Usage: Call SPRED2 (N, IP, X, Y, GR, GI, MAG, IPHA)

- N - Size of the array
- IP - Tape logical unit containing the denominator (PSF transform)
- X - Real part of the numerator (image transform) array
- Y - Imaginary part of the numerator array
- GR - Real part of the denominator (PSF transform) array
- GI - Imaginary part of denominator array
- MAG - Limit on magnitude amplification (integer)
 - blank - no limit
 - 0 - allow no magnitude change
 - other - limit all amplifications greater than RMAG to RMAG
- IPHA- Limit on phase amplification (integer)
 - blank - no limit
 - 0 - allow no phase change
 - other - limit all phase changes whose magnitude is greater than IPHA to IPHA

Method: The real/imaginary components are converted into magnitude/phase components.

Let A = image magnitude vector

- Θ = image phase vector
- B = PSF magnitude vector
- \emptyset = PSF phase vector

C = resulting magnitude vector

α = resulting phase vector

$$C/\alpha = \frac{A/\theta}{B/\phi} = \frac{A}{B} \angle \theta - \phi$$

If MAG = blank, operation above is carried out for all PSF points not equal to zero. If a PSF point is equal to zero, the corresponding point of the image magnitude vector is unaltered.

If MAG = 0; each point of B is set to one.

If MAG = other; for those points such that $1/B_i$ is greater than MAG, set $1/B_i$ to MAG.

If IPHA = blank; operation above is carried out completely.

If IPHA = 0; ϕ is set to zero.

If IPHA = other; if $|\phi|$ is greater than IPHA, ϕ is set to sign (ϕ) times IPHA.

Restrictions: (1) The arrays must be the same size and less than or equal to dimension 1024.

(2) The PSF transform vector is read from tape and must have been written in blocks of N real, N imaginary, N real, N imaginary, etc.

Storage: 474₍₈₎ words

Required by: DECONVOLVE

SPRED3

Purpose: SPRED3 is designed to perform multiplication of two complex vectors, and to compute the complex conjugate of the result.

Usage: Call SPRED3 (N, IP, X, Y, GR, GI)

N -- Size of the arrays

IP -- Tape logical unit containing PSF transform array.

X -- Real part of image transform array

Y -- Imaginary part of image transform array

GR -- Real part of PSF transform array

GI -- Imaginary part of PSF transform array

Method: Let $I = \text{image} = X + j Y$
 $P = \text{PSF} = GR + j GI$
 $O_R = \text{real part of resulting image and,}$
 $O_I = \text{imaginary part of resulting image.}$
then $(I)(P) = (X + jY) (GR + j GI) =$
 $(X)(GR) - (Y)(GI) + j [(Y)(GR) + (X)(GI)]$
 $O_R = (X)(GR) - (Y)(GI)$
 $O_I = -(Y)(GR) - (X)(GI)$

Restrictions: The arrays must be the same size and less than or equal to size 1024.

Array number two is read from tape and must have been written in blocks of N real, N imaginary, N real, N imaginary, etc.

Storage: $166_{(8)}$ words

Required by: CORRELATE

SHAVE (Frame Edge Rounding)

Purpose: SHAVE rounds off the edges of a frame of data.

Usage: CALL SHAVE (IN, IO, NL, NC, NP)

IN - Input tape logical unit number

IO - Output tape logical unit number

NL - Number of lines per frame

NC - Number of columns per frame

NP - Number of points in modification table

The modification table is input by SHAVE as follows:

Col.

1 *

MOD=(up to 25 points)\$

where MOD is a table of real values representing percentages by which the elements on the edge of the frame will be modified.

Storage: 4404₍₈₎ words

Required by: FEATHER

FLTOFX (Floating to Fixed Point Conversion)

Purpose: FLTOFX converts a frame of data from floating to fixed point format.

Usage: CALL FLTOFX (IN, IO, NL, NC)

IN - Input tape logical unit number

IO - Output tape logical unit number

NL - Number of lines per frame

NC - Number of columns per frame

Method: Each element within the frame is rounded to the nearest whole number and is output in integer format.

Storage: 10160₍₈₎ words

Required by: FLOAT TO FIXED

FXTOFL (Fixed to Floating Point Conversion)

Purpose: FXTOFL converts a frame of data from fixed to floating point format.

Usage: CALL FXTOFL (IN, IO, NL, NC)

IN - Input tape logical unit number

IO - Output tape logical unit number

NL - Number of lines per frame

NC - Number of columns per frame

Storage: 10162₍₈₎ words

Required by: FIXED TO FLOAT

STKPIC (Stack Frames of Data on Tape)

Purpose: STKPIC transfers a frame of data from one tape and stacks in on another as the next file.

Usage: CALL STKPIC (IN, IO, NL, NC)

IN - Input tape logical unit number

IO - Output tape logical unit number

NL - Number of lines per frame

NC - Number of columns per frame

Note: The output tape, IO, is not rewound.

Storage: 1131₍₈₎ words

Required by: STACK

MULDSP (Multiple Display Format)

Purpose: MULDSP arranges multiple pictures on a single frame in display format.

Usage: CALL MULDSP (IN, IO, IS, NL, NC, NX, NY)

IN - Input tape logical unit number

IO - Output tape logical unit number

IS - Scratch tape logical unit number

NL - Number of lines per input frame

NC - Number of columns per input frame

NX - Number of pictures to be put on one output frame.

If NX is positive, all pictures will be taken from successive files of the input tape. If negative, program will pause prior to processing each input picture to allow operator to change tapes, if necessary.

NY - Number of pictures across frame. If NY is positive, pictures will be separated by white space. If negative, pictures will be butted together.

Method: Successive pictures are taken from the input tape and formatted into a single 1024 x 1024 frame in display format.

Restrictions: (1) All input pictures must be the same size.
(2) Maximum input frame size is 512 x 512.
(3) Output frame size is 1024 x 1024.

Storage: 2305₍₈₎ words

Required by: MULTIPLE DISPLAY

LABL (Label)

Purpose: LABL inserts labels into a frame of data.

Usage: CALL LABL (IN, IO, NL, NP)

IN - Input tape

IO - Output tape

NL - Number of lines of data in input and output frame

NP - Number of lines of labels to be inserted

Method: A packed display format tape is read down to the specified line at which the label is to be inserted. The label is inserted and the process repeated until all labels have been inserted. The format of the label specification is:

Col 6	Col 12	Col 18	Col 25
J	K	L	(Any desired label)

J - Number of Characters in the label

K - Starting line position on the frame

L - Starting column position

The label begins in column 25 and continues for J columns. If the label is longer than 48 characters, it is continued on a second card beginning in column 1.

Each character is a 9x7 matrix with a 9x3 matrix of trailing blanks; therefore, each character occupies 9 lines and 10 columns. Line spacing is controlled by the user - a minimum of 5 lines between labels is recommended.

The character set is:

0 through 9, = ' - (). , * /, A through Z

The label is added to the original picture so that the labels essentially overlay the picture thus allowing overprinting, if desired.

15 March 1972

3-55

TM-(L)-HU-033/006/00

Restrictions: (1) The maximum number of characters per line is 102
(2) Col 72 of the label card is the last column interpreted. If the label extends beyond col 72, it must be continued on a second card.
(3) The starting line position specified on each successive label card must be at least 9 greater than the line position specification of the previous label card. That is, label lines may not overlap.

Subroutines: DSPLIN

Storage: 6724₍₈₎ words

Required by: LABEL

DSPLIN (Display Line)

Purpose: DSPLIN converts a string of BCD characters into a 9x171 array for display of alpha numeric characters on the Photowrite or DICOMED display.

Usage: CALL DSPLIN (IN, NC, KC, IO)

IN - String of characters to be converted to display format

NC - Number of characters in string to be converted counting all except trailing blanks

KC - Starting column position in output array

IO - Output array (9x171 packed format)

Method: Each BCD character in the input string is interpreted and a corresponding 9x7 bit array is placed in the output array.

Restrictions: A maximum of 102 characters may be displayed on one line.

Storage: 22566₍₈₎ words

Required by: MULDSP

CENTR (Center)

Purpose: CENTR multiplies each point of an input array by $(-1)^{i+j}$ where i is the line position of the point and j is its column position.

Usage: CALL CENTR (IN, IO, NL, NC, K)

IN - Input tape logical unit number
IO - Output tape logical unit number
NL - Number of lines in the image
NC - Number of columns in the image
K - Fixed/floating point flag
 K = 1 Perform the operation with fixed point arithmetic
 ≠ 1 Perform the operation with floating point arithmetic

Restrictions: NC must be less than or equal to 1024.

Storage: 2235₈ words

Required by: CENTER

SPLT (SPLIT DATA)

Purpose: SPLT splits complex data which is interleaved line-by-line onto two output tapes -- one containing the real (magnitude) component and the other containing the imaginary (phase) component. These are written line at a time. The entry MERG allows the separated complex data to be merged (interleaved) line by line onto one tape. An off-line print option is also included for the split operator.

Usage: For data separation

CALL SPLT (IN, Ø1, Ø2, NL, NC, NP)

For data combination

CALL MERGE (I1, I2, IO, NL, NC)

IN - Input tape logical unit number

Ø1 - Output tape logical unit number - magnitude or real data

Ø2 - Output tape logical unit number - phase or imaginary data

NL - Number of lines

NC - Number of columns

NP - Print flag

NP = 0 No print

= 1 Print magnitude or real (odd records)

= 2 Print phase or imaginary (even records)

= 3 Print all values

Restrictions: NC must be less than or equal to 1024.

Storage: 2573₍₈₎ words

Required by: SPLIT
MERGE

15 March 1972

3-59

TM-(L)-HU-033/006/00

MERG (Merge Data)

(See SPLT Subroutine write-up)

MPTORI (M/P to R/I)

Purpose: MPTORI converts a magnitude/phase complex tape (interleaved line by line) to a real/imaginary complex tape (interleaved line by line). The entry RITOMP converts from a R/I tape to a M/P tape.

Usage: To convert from M/P to R/I
CALL MPTORI (IN, IO, NL, NC)
To convert from R/I to M/P
CALL RITOMP (IN, IO, NL, NC)
IN - Input tape logical unit number
IO - Output tape logical unit number
NL - Number of lines
NC - Number of columns

Restrictions: NC must be less than or equal to 1024.

Storage: 10577₍₈₎ words

Required by: M/P to R/I, R/I to M/P

15 March 1972

3-61

TM-(L)-HU-033/006/00

RITOMP (R/I TO M/P)

(See MPTORI Subroutine write-up)

LOGMAG (Display Transforms)

Purpose: LOGMAG converts a complex (magnitude/phase) line by line interleaved tape of the transform of an image into two images (magnitude, phase), each quantized to less than or equal to 64 levels in order that they can be viewed on the DICOMED display.

Usage: CALL LOGMAG (NL, NC, IN, I01, I02)

NL - Number of lines
NC - Number of columns
IN - Input tape logical unit number
I01 - Output tape logical unit number - magnitude
I02 - Output tape logical unit number - phase

Method: In order to display the magnitude of the Fourier transform of an image, the values must be somehow reduced to less than or equal to 64 levels. The natural logarithm is first taken of the magnitude term to reduce the range. Then the maximum and minimum values of this logarithm are determined. The spread between these maximum and minimum values is then uniformly quantized to 64 levels such that zero represents the smallest magnitude. The integer output is written on unit I01.

Since the phase of the transform can vary between a limited range (-180° to $+180^\circ$), this spread was uniformly quantized directly to 60 levels where -180° is represented by 0, zero degrees is represented by 30, and $+180^\circ$ by 60. The integer output is written on unit I02.

Restrictions: NC must be less than or equal to 256.

Storage: 2543₍₈₎ words

Required by: LOG MAGNITUDE

PDAFNL (Position Dependent Alteration - Floating Point - No Limiting)

Purpose: PDAFNL modifies each element of a frame of data as a function of its position within the frame.

Usage: CALL PDAFNL (IN, IO, NL, NC, NP)

IN - Input tape logical unit number

IO - Output tape logical unit number

NL - Number of lines

NC - Number of columns

NP - Number of points in the modification tables.

The modification tables are input by PDAFNL as follows:

Col.

1*

A = (up to 2048 values)\$

B = (up to 2048 values)\$

C = (up to 2048 values)\$

The values of tables A, B, and C represent the coefficients of the expansion $(A(M)+B(M)X+C(M)X^2)PIC(i,j)$

Method: The radial distance of an element from the center of the frame is computed and is used as the index (M) to tables A, B, and C in calculating a modified value as

$$\text{Modified value} = (A(M)+B(M)X+C(M)X^2)PIC(i,j)$$

where $X=U+V$: $U=\omega x$ expressed in cycles/frame integer,

$V=\omega y$ cycles/frame integer

$PIC(i,j)$ =original value of the (i,j) picture element.

*Continuation cards begin in column #2.

15 March 1972

3-64

TM-(L)-HU-033/006/00

Restrictions: (1) Tables A, B, and C may not exceed 2048 values.

(2) All alterations are symetrical about the center point in the frame of data.

Storage: 20372₍₈₎ words

Required by: GRADIENT

15 March 1972

3-65

TM-(L)-HU-033/006/00

TCOPY (Tape Copy)

Purpose: TCOPY copies a file from an input tape to a file of an output tape.

Usage: CALL TCOPY (IN, IO, IF, OF, KF, NL, NC)

IN - Input tape

IO - Output tape

IF - File on input tape

OF - File on output tape

KF - Format flag

KF = 1 - Interleaved

KF = 2 - Scanner format

KF = other - Not interleaved, FORTRAN format

NL - Number of lines

NC - Number of columns

Storage: 4360₈ words

Required by: COPY

ARITH (Arithmetic Functions)

Purpose: ARITH performs the following point by point operations on two frames of data: $A+B$, $A-B$, $A*B$, A/B . Also $A+C$ and $A*C$, where C is a constant, may be performed.

Usage: CALL ARITH (I1, I2, IO, NL, NC, KF, LF, CN)

- I1 - Input tape (frame A)
- I2 - Input tape (frame B)
- IO - Output tape
- NL - Number of lines
- NC - Number of columns
- KF - Operation flag
 - KF = 1 - $A+B$
 - KF = 2 - $A-B$
 - KF = 3 - $A*B$
 - KF = 4 - A/B
 - KF = 5 - $A+C$
 - KF = 6 - $A*C$
- LF - Data type flag
 - LF = 1 - Integer (FORTRAN unpacked format)
 - LF = 2 - Integer (Display packed format)
 - LF = Other - Floating point
- CN - Constant to be used if KF = 5 or 6 (floating point)

Storage: 17121₍₈₎ words

Required by: MATH

SCAL (Scale)

Purpose: SCAL scales a frame of data to values between 0. and 63.

Usage: CALL SCAL (IN, IO, IS, NL, NC, KF, LF, BC, TC)

IN - Input tape

IO - Output tape

IS - Scratch tape

NL - Number of lines

NC - Number of columns

KF - Curve flag

KF = 1 - logarithmic (base e)

KF = 2 - square root

KF = 3 - cube root

KF = 4 - square

KF = other - linear

LF - Clipping flag

LF = 1 - Clip outside of BC and TC

LF = other - No clipping

BC - Lower clipping level - all values below BC will be set to 0.

TC - Upper clipping level - all values above TC will be set to 63.

Method: The minimum and maximum value in the frame of data is determined by searching through the frame.

The spread is computed as

$$SP = MAX - MIN$$

and the step size, or increment, is

$$ST = SP/64.$$

The tape is then rescanned to scale each element between 0. and 63. by the desired curve.

Storage: 4623₍₈₎ words

Required by: SCALE